FRAMES2LORA: Parametric Video Internalization for Vision-Language Models

Manan Suri[†*], **Sarvesh Baskar**[*], **Dinesh Manocha**[†]
[†]University of Maryland, College Park
{manans,dmanocha}@umd.edu baskarsarvesh@gmail.com

Abstract

Processing video in vision-language models is expensive: each frame occupies hundreds of tokens, and inference cost scales with every frame and every repeated query. We introduce **FRAMES2LORA**, a method for parametric video internalization. A perceiver hypernetwork reads the intermediate representations produced layer-by-layer as a frozen VLM encodes a video, and generates a Low-Rank Adaptation (LoRA) adapter in a single forward pass. Unlike standard LoRA fine-tuning, which requires iterative gradient updates, FRAMES2LORA predicts these weights directly from the video. Trained for SmolVLM2 500M and 2.2B on video summarization and captioning, FRAMES2LORA enables the same frozen VLM to answer queries from the adapter alone, with ***zero visual tokens*** in its context at query time. FRAMES2LORA is statistically non-inferior and equivalent to direct video-in-context inference across all five captioning benchmarks at both model scales, and across seven of eight video question answering benchmark-scale pairings. Although trained only on 12 frames at 384px, it remains stable up to 1,024 frames and 1024px, where direct video-in-context inference often degenerates. Across this sweep, it reduces answer-time visual-token load by up to 1,500× and query TTFT by 6–80×, while preserving video-faithful outputs. We also find that independently generated adapters for non-overlapping video segments can compose in rank space, suggesting a path toward chunked long-video internalization.

1 Introduction

Video understanding in VLMs is built on a token-heavy abstraction: frames are encoded as visual tokens and concatenated into the model's context window. Each frame at standard resolution contributes hundreds of visual tokens (Liu et al., 2024; Shang et al., 2025); even short clips of a few dozen frames generate tens of thousands of tokens before any text query is added, and memory and latency scale with every frame and every query. Past a capacity threshold, this bottleneck does not produce gracefully degraded outputs: VLMs generate incoherent or repetitive text unrelated to the video (Chen et al., 2025b; Zhang et al., 2024). The context window (the model's fixed token capacity) is therefore the fundamental bottleneck for video understanding, and it is re-encountered on every query over the same video.

Much work aims to fit more video into the context window. Frame subsampling (Zhang et al., 2023) discards frames to meet a token budget, sacrificing temporal coverage. Visual token compression methods (Shang et al., 2025; Li et al., 2025) prune or merge spatial tokens before the language backbone, reducing per-frame cost without discarding entire frames. Long-context architectures (Chen et al., 2025b; Zhang et al., 2024) scale the context window itself through sequence parallelism and position encoding modifications. Streaming methods (Qian et al., 2024) process video incrementally, maintaining a compact memory buffer in lieu of full context retention. Each approach reduces the burden without resolving it: visual tokens remain in context at query time, every query re-incurs the encoding overhead, and all approaches eventually encounter the same capacity ceiling. The capacity ceiling is not a constraint to manage: it is a constraint to eliminate.

We take a fundamentally different approach. Rather than compressing visual information to fit within the context window, we eliminate it from the query entirely, encoding the video into the model's parameters before any query is issued. The video is stored as a LoRA adapter (Hu et al., 2022); subsequent queries are answered by a frozen base model with those adapter weights, with no visual tokens in context. Prior work has shown that feedforward hypernetworks (Ha et al., 2017; Charakorn et al., 2026) can produce LoRA adapters from *text documents*, enabling a frozen LLM to answer queries about a document with no text tokens in context. Extending this paradigm to video introduces qualitatively harder challenges: the token volume per example is orders of magnitude larger, making iterative per-example optimization computationally

[*]Equal contribution.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 34951–34989
October 24-29, 2026 ©2026 Association for Computational Linguistics

impractical; the compression is cross-modal, requiring visual semantics to be expressed as perturbations to a language model's parameter space; and the visual distribution varies along a resolution axis with no textual analog.

Main Result: We introduce **FRAMES2LORA**, a framework for parametrically internalizing videos into a frozen vision-language model (VLM). Given a video, a perceiver hypernetwork (Jaegle et al., 2021) processes the layer-wise hidden states of the frozen VLM encoder and generates LoRA adapter weights in a single forward pass. The generated adapter is then attached to the same frozen VLM, enabling it to answer questions about the video without requiring visual tokens in the context window. During training, both the VLM encoder and the answering model remain frozen; only the hypernetwork is optimized using cached teacher-generated captions and summaries as supervision. We train and evaluate FRAMES2LORA on SmolVLM2 500M and 2.2B (Marafioti et al., 2025). Our novel contributions include:

- **First parametric video internalization.** A Perceiver hypernetwork that converts a video into a LoRA adapter in a single forward pass, enabling a frozen VLM to answer queries with no visual tokens in context. We demonstrate feasibility across 2.2B and 500M model scales.
- **Strong performance on captioning and video question answering.** Statistical non-inferiority and equivalence to direct video-in-context inference across all five captioning benchmarks at both model scales (ActivityNet Captions, PLM-RDCap, PLM-RCap, VDC, CaReBench) and across seven of eight video question answering benchmark-scale pairings (NExT-QA, ActivityNet-QA, PLM-SGQA, VidCapBench).
- **Efficiency, generalization and emergent compositionality.** Although trained only on 12 frames at 384px, FRAMES2LORA remains stable up to 1,024 frames and 1024px, where direct video-in-context inference often degenerates. It reduces answer-time visual-token load by up to 1,500× and query TTFT by 6–80×, while preserving video-faithful outputs. Compared to KV caching and token-compression techniques, we show that video internalization via FRAMES2LORA preserves performance across token budgets, is faster to process, and has the lowest time to first token. We further observe that adapters generated independently for non-overlapping video segments can compose in rank space, suggesting a path toward chunked long-video internalization.

2 Related Work

2.1 Efficient Video Understanding

Most efficient video-understanding methods reduce the number or cost of visual tokens while still keeping visual information in the model context. Frame subsampling (Zhang et al., 2023) lowers temporal coverage to fit a token budget; visual-token compression (Shang et al., 2025; Li et al., 2025) prunes or merges spatial tokens; long-context video models (Chen et al., 2025b; Zhang et al., 2024) extend the usable context window; and streaming methods (Qian et al., 2024; Zhang et al., 2023) maintain compact memory across time. These approaches improve scalability, but the language model still conditions on visual tokens at query time. FRAMES2LORA is orthogonal: it converts the video into adapter weights once, then answers later queries without visual tokens in context.

2.2 Parametric Knowledge Compression

Parameter-efficient methods such as LoRA, prefix tuning, and prompt tuning store task information in small learned updates rather than full model parameters (Hu et al., 2022; Li and Liang, 2021; Lester et al., 2021). More recent work moves instance-level context into compact representations, including gist tokens (Mu et al., 2023), hypernetwork-based editing (Mitchell et al., 2022; Ha et al., 2017), and deep context distillation (Caccia et al., 2025). Closest to our setting, Doc-to-LoRA maps text documents into LoRA adapters using a feedforward hypernetwork (Charakorn et al., 2026), while ViPE (Lu et al., 2025) explores visual perception in parameter space. FRAMES2LORA extends this idea from text to video, where the hypernetwork must compress high-volume visual context into language-model adapter weights and generalize across frame count and resolution.

3 FRAMES2LORA

FRAMES2LORA converts a video into a video-specific LoRA adapter in a single forward pass. A

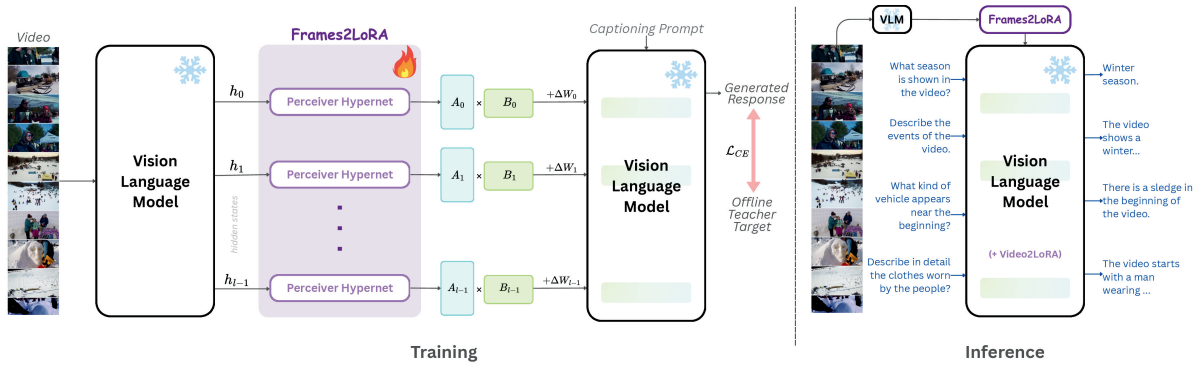


Figure 1: **FRAMES2LORA overview.** *Training (left):* A frozen VLM encodes the input video into hidden states. The trainable FRAMES2LORA hypernetwork reads these states and generates LoRA adapter weights in a single forward pass. The adapter-augmented frozen VLM is trained against teacher-generated targets. *Inference (right):* Given a new video, FRAMES2LORA generates the LoRA adapter once. The frozen VLM, augmented with this adapter, answers arbitrary text queries without visual tokens. Per-query cost is independent of video length.

frozen VLM encodes the video into layer-wise hidden states, and a trainable Perceiver hypernetwork maps these states into LoRA weights. At inference time, the generated adapter is attached to the frozen answer model, which answers downstream text prompts without receiving any visual tokens in its context.

3.1 Problem Formulation

Let v denote a video, i an internalization instruction, p a downstream text prompt, and y the target response. We assume a frozen vision-language encoder E, a frozen answer model F, and a trainable hypernetwork H_ϕ. The method is defined as:

$$\mathbf{C} = E(v, i), \tag{1}$$

$$\theta(v) = H_\phi(\mathbf{C}), \tag{2}$$

$$p_\phi(y \mid p, v) = F(y \mid p; \theta(v)). \tag{3}$$

Here, $\mathbf{C}$ denotes video-conditioned hidden states and $\theta(v)$ denotes the generated LoRA adapter. The answer model receives the text prompt p and the adapter $\theta(v)$, but not the video tokens. During training, only ϕ is updated; both E and F remain frozen.

3.2 Video Encoder

We use a frozen SmolVLM2 model (Marafioti et al., 2025) as the video encoder. Given a sampled video and the internalization instruction, we collect the text-side hidden states from each transformer layer:

$$\mathbf{C} = \mathrm{stack}(\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{L-1}) \in \mathbb{R}^{L \times S \times D}, \tag{4}$$

where L is the number of layers, S is the fused sequence length, and D is the hidden dimension. Keeping the layer dimension allows the hypernetwork to generate layer-indexed adapters instead of using a single pooled video vector for all layers.

3.3 Perceiver Hypernetwork

The hypernetwork maps $\mathbf{C}$ into LoRA weights for selected linear modules of the frozen model. We use a Perceiver-style resampler architecture (Jaegle et al., 2021). For each layer slice $\mathbf{C}_\ell \in \mathbb{R}^{S \times D}$, an encoder resampler attends from learned latent queries to the video-conditioned hidden states, producing a fixed-size representation. A decoder resampler then uses one output query for each target module and LoRA rank direction.

For batch size B, number of target modules M, rank R, and latent size Z, the hypernetwork output has shape

$$\mathbf{O} \in \mathbb{R}^{B \times L \times M \times R \times Z}. \tag{5}$$

A shared projection head maps each rank latent to the two LoRA factors:

$$\begin{aligned} \mathbf{A}_{\ell,m} &\in \mathbb{R}^{R \times d_{\text{in}}}, \\ \mathbf{B}_{\ell,m} &\in \mathbb{R}^{R \times d_{\text{out}}}. \end{aligned} \tag{6}$$

where ℓ indexes the transformer layer and m indexes the target linear module. The generated factors are scaled by learned multipliers, with the $\mathbf{A}$ scale initialized to one and the $\mathbf{B}$ scale initialized to zero.

3.4 Dynamic LoRA Injection

For a frozen linear layer with weight $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, we use the standard LoRA factorization (Hu et al., 2022). Under the row-vector implementation convention, the frozen layer computes

$\mathbf{x}\mathbf{W}^\top$. The generated adapter adds:

$$\Delta\mathbf{y} = s\,(\mathbf{x}\mathbf{A}_{\ell,m}^\top)\mathbf{B}_{\ell,m}, \tag{7}$$

where s is the fixed LoRA scaling factor. The full adapted forward pass is:

$$\mathbf{y} = \mathbf{x}\mathbf{W}^\top + s\,(\mathbf{x}\mathbf{A}_{\ell,m}^\top)\mathbf{B}_{\ell,m}. \tag{8}$$

Each example receives its own generated adapter, so the LoRA weights are conditioned on the input video rather than shared across all videos.

3.5 Training Objective

We train the hypernetwork with teacher-forced cross-entropy over response tokens:

$$\mathcal{L}(\phi) = -\sum_t \log p_\phi(y_t \mid y_{<t}, p, \theta(v)). \tag{9}$$

The answer model receives only the downstream text prompt and the generated adapter during this loss computation.

4 Experimental Setup

4.1 Models and Training

We evaluate two SmolVLM2 model scales: 500M and 2.2B. For each scale, the video encoder and answer model are initialized from the same frozen backbone. Only the FRAMES2LORA hypernetwork is trained. Training uses 12 uniformly sampled frames at 384px longest-edge resolution (constrained by compute). We apply generated LoRA adapters to the MLP `down_proj` modules of the text decoder, with rank $R = 16$. We train on video spans derived from FineVideo (Farré et al., 2024). The span mixture contains single-scene spans, adjacent multi-scene spans, and full-video spans, sampled in a 60/30/10 ratio. FineVideo metadata is used only to define spans; the final training targets are cached offline teacher generations from a frozen SmolVLM2 teacher conditioned on the sampled video frames and downstream prompt. Audio is excluded throughout. The hypernetwork is trained with teacher-forced cross-entropy over response tokens, while the answer model receives only the text prompt and generated adapter. Further details on training can be found in the appendix.

4.2 Evaluation Benchmarks

We evaluate captioning on ActivityNet Captions (Krishna et al., 2017), PLM-RDCap (Cho et al., 2025), PLM-RCap (Cho et al., 2025), VDC (Chai et al., 2025), and CaReBench (Xu et al., 2025); and video QA on NExT-QA (Xiao et al., 2021), ActivityNet-QA (Yu et al., 2019), PLM-SGQA (Cho et al., 2025), and VidCapBench (Chen et al., 2025a).

To scale LLM Judge evaluation, we fix the number of samples from each benchmark to 500. VDC and CaReBench use 500 examples per subset/style. VidCapBench has multiple QA pairs corresponding to each video, therefore we fixed the number of videos to 100 and obtained 1,523 QA pairs corresponding to it. For all benchmarks, the direct baseline and FRAMES2LORA use the same videos, prompts, references, frame sampling, and decoding configuration.

4.3 Metrics and Statistical Testing

We report two quality metrics. First, we compute token-level F1 between the generated output and the reference answer or caption. Second, we use an LLM judge to score output quality on a 1–5 scale, which is linearly rescaled to $[0, 1]$. We use Qwen3-30B (Yang et al., 2025) as our judge model, with a constrained rubric. Human study on this metric for a subset of 200 examples (100 captioning + 100 QA) reveals strong correlation with human judgements, with Spearman $\rho = 0.823$ for metric fidelity.

We estimate 95% confidence intervals using paired bootstrap resampling. For statistical measures, *NI* (Non-inferiority) and *Eq* (Equivalence) we use a margin of 0.05 for token-F1 and 0.15 for rescaled judge score.

5 Results

5.1 Captioning

FRAMES2LORA passes both non-inferiority and equivalence on **all 10 benchmark–scale combinations** under the LLM judge and all 10 under token-F1 (Table 1). For SmolVLM 2.2B, FRAMES2LORA recovers **91.9%** of the base model's judge score, while for SmolVLM 500M, it recovers **84.2%** (see Appendix F.1 for video-grounded VLM judging).

Per-benchmark analysis. Recovery rates at 500M span 79–96%, with compact clip-aligned benchmarks (PLM-RCap, PLM-RDCap) easiest to internalize and temporally dense benchmarks (VDC, ActivityNet Captions) hardest. Scale narrows this spread considerably: at 2.2B the floor rises to 85% and the ceiling breaks above the base,

LLM Judge

Benchmark	SmolVLM 500M						SmolVLM 2.2B					
	Base	V2L	Δ	CI	Eq	NI	Base	V2L	Δ	CI	Eq	NI
ActivityNet Captions	0.428	0.356	-0.072	[-0.104, -0.041]	**Y**	**Y**	0.576	0.492	-0.084	[-0.113, -0.057]	**Y**	**Y**
PLM-RDCap	0.308	0.263	-0.045	[-0.069, -0.021]	**Y**	**Y**	0.326	0.316	-0.010	[-0.032, +0.012]	**Y**	**Y**
PLM-RCap	0.252	0.242	-0.011	[-0.031, +0.009]	**Y**	**Y**	0.270	0.287	+0.017	[+0.001, +0.034]	**Y**	**Y**
VDC (aggregate)	0.515	0.406	-0.108	[-0.118, -0.098]	**Y**	**Y**	0.539	0.511	-0.028	[-0.037, -0.019]	**Y**	**Y**
CaReBench	0.334	0.278	-0.056	[-0.067, -0.045]	**Y**	**Y**	0.437	0.369	-0.068	[-0.078, -0.058]	**Y**	**Y**
Average	0.367	0.309	-0.058	[-0.078, -0.039]	**Y**	**Y**	0.430	0.395	-0.035	[-0.052, -0.018]	**Y**	**Y**

Token F1

Benchmark	SmolVLM 500M						SmolVLM 2.2B					
	Base	V2L	Δ	CI	Eq	NI	Base	V2L	Δ	CI	Eq	NI
ActivityNet Captions	0.236	0.243	+0.007	[+0.002, +0.012]	**Y**	**Y**	0.263	0.256	-0.007	[-0.012, -0.002]	**Y**	**Y**
PLM-RDCap	0.189	0.198	+0.009	[+0.005, +0.013]	**Y**	**Y**	0.198	0.207	+0.009	[+0.005, +0.013]	**Y**	**Y**
PLM-RCap	0.177	0.203	+0.026	[+0.021, +0.031]	**Y**	**Y**	0.199	0.204	+0.005	[+0.001, +0.010]	**Y**	**Y**
VDC (aggregate)	0.315	0.288	-0.027	[-0.030, -0.025]	**Y**	**Y**	0.297	0.304	+0.007	[+0.003, +0.010]	**Y**	**Y**
CaReBench	0.295	0.275	-0.020	[-0.023, -0.017]	**Y**	**Y**	0.292	0.279	-0.013	[-0.015, -0.010]	**Y**	**Y**
Average	0.243	0.242	-0.001	[-0.005, +0.003]	**Y**	**Y**	0.250	0.250	+0.000	[-0.004, +0.004]	**Y**	**Y**

Table 1: Comparison of the base model with video and FRAMES2LORA generated adapters, across captioning benchmarks using LLM Judge scores and Token F1. We report mean scores, the paired difference Δ (V2L − Base), 95% confidence intervals, and the statistical equivalence (Eq) and non-inferiority (NI) criteria.

Subset	SmolVLM 500M		SmolVLM 2.2B	
	Base	V2L (Δ)	Base	V2L (Δ)
Short caption	0.629	0.535 (-0.094)	0.556	0.579 (+0.022)
Detailed caption	0.476	0.401 (-0.074)	0.526	0.463 (-0.063)
Camera	0.310	0.131 (-0.178)	0.478	0.392 (-0.085)
Background	0.642	0.523 (-0.117)	0.588	0.606 (+0.018)
Main object	0.517	0.442 (-0.075)	0.546	0.514 (-0.032)

Table 2: VDC results broken down by caption style.

Subset	SmolVLM 500M		SmolVLM 2.2B	
	Base	V2L (Δ)	Base	V2L (Δ)
Caption	0.418	0.324 (-0.094)	0.465	0.400 (-0.065)
Events	0.201	0.169 (-0.032)	0.340	0.267 (-0.073)
Objects	0.368	0.327 (-0.043)	0.457	0.392 (-0.065)
Spatial caption	0.424	0.329 (-0.095)	0.519	0.426 (-0.094)
Temporal caption	0.260	0.242 (-0.018)	0.404	0.360 (-0.045)

Table 3: CaReBench results broken down by subset.

with PLM-RCap *surpassing* the base outright (CI entirely above zero) and PLM-RDCap reaching de-facto equivalence (CI straddling zero). The benchmarks most sensitive to scale—particularly VDC, where the gap contracts fourfold—are those requiring compression of visually diverse, longer-form descriptions; benchmarks with consistently structured references recover well at both scales.

Token F1. Token-F1 provides independent reference-based corroboration: the mean paired delta is -0.001 at 500M and 0.000 at 2.2B. FRAMES2LORA exceeds base on 3 of 5 benchmarks at 500M (ActivityNet Captions $+0.007$, PLM-RDCap $+0.009$, PLM-RCap $+0.026$) and 2 of 5 at 2.2B. The PLM-RCap result at 500M is notable: $+0.026$ ($+14.7\%$; CI $[+0.021, +0.031]$) with no token-level supervision.

5.2 Fine-Grained Captioning

Tables 2 and 3 break VDC and CaReBench into caption styles and semantic dimensions.

VDC Four of five VDC styles maintain 81–85% recovery at 500M: short (85.1%, $\Delta = -0.094$), detailed (84.2%, $\Delta = -0.074$), background (81.5%, $\Delta = -0.117$), main object (85.5%, $\Delta = -0.075$). *Camera* captions are the outlier: At 500M, FRAMES2LORA achieves only 42.3% recovery ($\Delta = -0.178$; base 0.310, V2L 0.131), as cinematographic attributes—shot framing, viewpoint, and camera motion—are difficult to encode as weight perturbations at this scale. At 2.2B, FRAMES2LORA recovers 82.0% ($\Delta = -0.085$), a gain of +39.7 pp. FRAMES2LORA recovers **82.0%** ($\Delta = -0.085$), a gain of **+39.7 pp**—the largest single-dimension scale improvement in the fine-grained evaluation—This suggests that part of the camera-description gap is capacity-related, although targeted camera-motion supervision or adaptive rank may still be needed. At 2.2B, two styles exceed the base outright: short captions (104.1%, $\Delta = +0.022$) and background (103.1%,

LLM Judge

Benchmark	SmolVLM 500M						SmolVLM 2.2B					
	Base	V2L	Δ	CI	Eq	NI	Base	V2L	Δ	CI	Eq	NI
NExT-QA (open)	0.501	0.547	+0.046	[+0.007, +0.084]	**Y**	**Y**	0.597	0.610	+0.013	[-0.022, +0.048]	**Y**	**Y**
ActivityNet-QA	0.524	0.541	+0.016	[-0.031, +0.064]	**Y**	**Y**	0.627	0.531	-0.096	[-0.144, -0.049]	**Y**	**Y**
PLM-SGQA	0.390	0.317	-0.074	[-0.113, -0.034]	**Y**	**Y**	0.493	0.295	-0.198	[-0.236, -0.161]	–	–
VidCapBench	0.502	0.451	-0.050	[-0.071, -0.030]	**Y**	**Y**	0.551	0.475	-0.076	[-0.096, -0.055]	**Y**	**Y**
Average	0.487	0.460	-0.027	[-0.043, -0.011]	**Y**	**Y**	0.562	0.477	-0.085	[-0.101, -0.069]	**Y**	**Y**

Token F1

Benchmark	SmolVLM 500M						SmolVLM 2.2B					
	Base	V2L	Δ	CI	Eq	NI	Base	V2L	Δ	CI	Eq	NI
NExT-QA (open)	0.129	0.068	-0.061	[-0.076, -0.046]	–	–	0.140	0.076	-0.063	[-0.079, -0.048]	–	–
ActivityNet-QA	0.197	0.023	-0.174	[-0.199, -0.149]	–	–	0.149	0.013	-0.136	[-0.156, -0.117]	–	–
PLM-SGQA	0.081	0.225	+0.145	[+0.131, +0.158]	–	**Y**	0.092	0.203	+0.111	[+0.098, +0.124]	–	**Y**
VidCapBench	0.216	0.209	-0.007	[-0.019, +0.004]	**Y**	**Y**	0.196	0.218	+0.022	[+0.010, +0.033]	**Y**	**Y**
Average	0.156	0.131	-0.024	[-0.041, -0.008]	**Y**	**Y**	0.144	0.128	-0.017	[-0.032, -0.002]	**Y**	**Y**

Table 4: Comparison of the base model with video and FRAMES2LORA generated adapters, across video question answering benchmarks using LLM Judge scores and Token F1. We report mean scores, the paired difference Δ (V2L − Base), 95% confidence intervals, and the statistical equivalence (Eq) and non-inferiority (NI) criteria.

$\Delta = +0.018$).

CaReBench Temporal captioning is best-recovered at both scales (500M: 93.1%, $\Delta = -0.018$; 2.2B: 89.1%, $\Delta = -0.045$); objects follow (500M: 88.9%; 2.2B: 85.8%). Holistic captioning and spatial description are hardest at 500M (77.5% and 77.6%), but scale closes the gap strongly: holistic reaches 86.0% (+8.5 pp) and spatial 82.1% (+4.5 pp) at 2.2B. The events dimension inverts: recovery falls from 84.1% (500M) to 78.5% (2.2B, −5.6 pp) as the 2.2B base improves substantially on event enumeration (base: 0.201 → 0.340), raising the compression target beyond the adapter's fixed rank.

5.3 Video Question Answering

FRAMES2LORA is trained exclusively on captioning; video QA is entirely a zero-shot transfer task. The LLM judge passes non-inferiority and equivalence on **7 of 8** benchmark–scale combinations (Table 4).

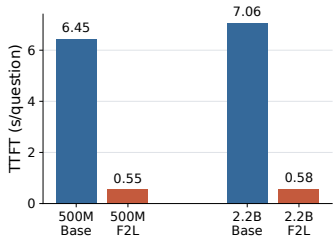
Per-benchmark judge analysis. Across the four QA benchmarks, FRAMES2LORA matches or exceeds the base on two of four at 500M and one of four at 2.2B, with NExT-QA being the standout: FRAMES2LORA *surpasses* the base at both scales, with the 500M CI lying entirely above zero. The single failure—PLM-SGQA at 2.2B—is instructive rather than representative. The same benchmark passes comfortably at 500M points does not point to a fundamental limitation of parametric QA internalization.

Token-F1 and the verbosity effect. Token-F1 diverges from the judge on short-answer QA due to format mismatch. On ActivityNet-QA, references average only 1.3 words while FRAMES2LORA outputs average 17.7–21.4 words (14–17× longer); similarly, on NExT-QA, references average 2.7 words vs. 17.1–18.0 words for FRAMES2LORA (6.3–6.6× longer). This length inflation sharply lowers unigram Token-F1 despite correct semantic content. Conversely, on PLM-SGQA where references are descriptive (21.9 words), FRAMES2LORA (23.8–27.8 words) achieves a positive delta (500M: $\Delta = +0.145$; 2.2B: $\Delta = +0.111$), and VidCapBench reaches near-parity ($\Delta = -0.007$ / $+0.022$).

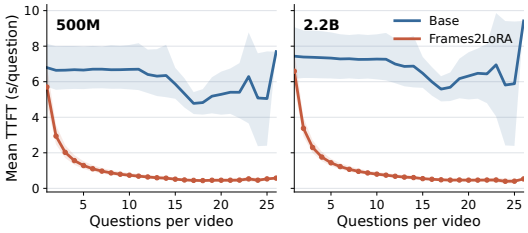
5.4 Frame and Resolution Generalization

FRAMES2LORA checkpoints were trained with uniform sampling at 12 frames and 384px resolution. We test out-of-distribution scaling on VDC background captioning by sweeping $\{8, 12, 24, 48, 128, 256, 512, 1024\}$ frames and $\{224, 336, 512, 1024\}$ resolution for both 500M and 2.2B models. We compare video-in-context inference with FRAMES2LORA using Token-F1, query-time TTFT (Time to First Token), and input-token reduction during answering (Fig. 3).

Despite being trained at a single frame count-

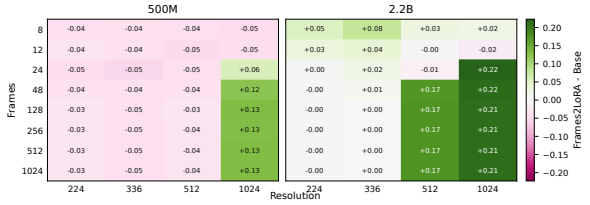


(a) Single-question average TTFT, with time taken to internalize the video accounted.

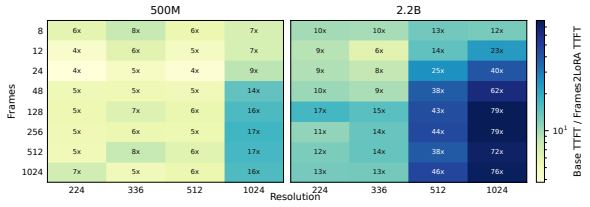


(b) Amortized TTFT per question vs. number of questions per video (shaded band = bootstrap 95% confidence interval).

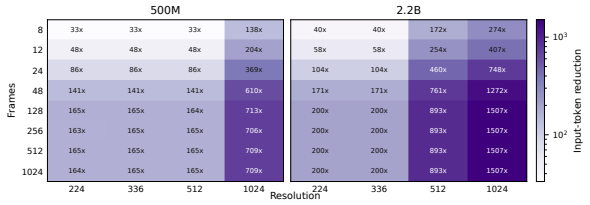
Figure 2: Inference efficiency on VidCapBench, comparing the base model and FRAMES2LORA.



(a) Change in mean Token-F1 from replacing in-context video tokens with FRAMES2LORA.



(b) Query-time TTFT speedup of FRAMES2LORA over the base video-in-context model.



(c) Input-token reduction achieved by FRAMES2LORA during answering.

Figure 3: Scaling behavior on VDC background captioning across frame count and spatial resolution.

resolution setting, FRAMES2LORA remains stable across the sweep. For 500M, FRAMES2LORA is close to the base model overall, with an average Token-F1 change of -0.012. At 1024px and high frame counts, however, FRAMES2LORA outperforms the base model by $+0.12$ to $+0.13$ Token-F1. This large gain is partly because direct video-in-context inference becomes unstable in this regime: the base model often degenerates into repetitive or gibberish generations when significantly large number of visual tokens are supplied. We also evaluate on long-form video under LONGVIDEOBENCH (Appendix F.2). The efficiency gains grow with video scale. FRAMES2LORA reduces query TTFT by a geometric mean of $6.7\times$ for 500M and $20.1\times$ for 2.2B, with maximum speedups of $17.2\times$ and $79.1\times$, respectively (Fig. 3b). This is explained by the token compression in Fig. 3c: FRAMES2LORA reduces answer-time input tokens by $150\times$ for 500M and $302\times$ for 2.2B on average, reaching $713\times$ and $1507\times$ at the largest settings, since it passes **zero tokens** during inference.

5.5 Inference Efficiency

VidCapBench is a natural setting for evaluating inference efficiency because each video is associated with multiple questions: in our evaluation split, 100 videos produce 1,523 total queries, or 15.23 questions per video on average. This matches the intended use case of FRAMES2LORA: the video is processed once to produce a video-specific LoRA, and the adapter is reused for all subsequent questions about the same video. Thus, unlike direct in-context video inference, which repeatedly pays the cost of encoding and conditioning on the video, FRAMES2LORA pays a one-time setup cost and amortizes it over repeated queries. Figure 2 shows this amortization effect on both the 500M and 2.2B backbones. Averaged over all VidCapBench queries, FRAMES2LORA reduces TTFT from 6.45s to 0.55s for the 500M model, an $11.75\times$ speedup, and from 7.06s to 0.58s for the 2.2B model, a $12.11\times$ speedup (Figure 2a). The prefix-amortization curve in Figure 2b shows that after 5 questions, amortized TTFT drops to 1.29s for 500M and 1.44s for 2.2B; after 10 questions, it falls to 0.74s and 0.80s, respectively.

Figure 4 further studies different video inference strategies on 640 samples with varying token counts (by doing using a resolution, frame count grid). We compare FRAMES2LORA and the default setting with, FrameFusion (Fu et al., 2025) (a token compression technique, compression fac-

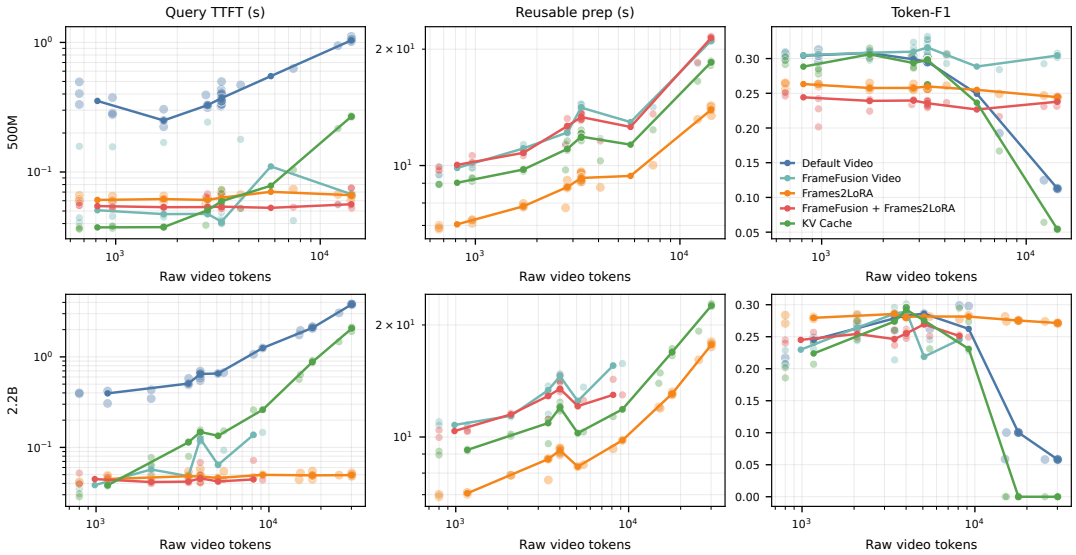


Figure 4: Efficiency comparison across video-token budgets. Columns report query TTFT, reusable preprocessing cost (internalization for FRAMES2LORA, cache creation for KV Cache, and token compression for FrameFusion), and Token-F1.

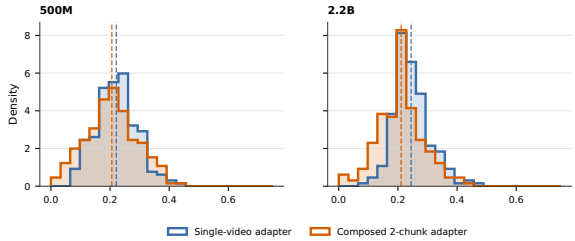


Figure 5: Two-chunk adapter composition on VDC.

tor 4), and KV caching. We also use FrameFusion with FRAMES2LORA, to show FRAMES2LORA is compatible with existing token compression techniques. Across token budgets, FRAMES2LORA is the only method that provides all three properties needed for repeated video querying: (1) query TTFT stays nearly constant and low as video tokens grow, (2) reusable preparation is competitive or fastest and much cheaper than KV caching at scale, and (3) output quality remains stable as token count increases. In contrast, the default baseline, token compression results and KV caching scale with token counts. Together, these results show that FRAMES2LORA converts video conditioning from a repeated per-query overhead into a reusable video-specific computation.

5.6 Chunk Composition

FRAMES2LORA internalizes a video by generating a LoRA adapter from its visual context. Although the model is trained to produce adapters for single video contexts, the adapter representation admits a simple test-time composition operation: independently internalize two temporal chunks of the same video, concatenate the resulting LoRA ranks, and decode from the composed adapter. We evaluate whether this operation produces coherent video-level generations, rather than degenerate text or captions tied to only one chunk.

We use the VDC short-caption and detailed-caption subsets, with 100 videos from each subset. Each video is split into two equal temporal halves. We compare two conditions: *single-video adapter*, where the full video is internalized as one adapter, and *composed two-chunk adapter*, where the two halves are internalized independently and the resulting adapters are composed before generation. Both conditions use 12 frames per adapter and the same text prompt. Figure 5 shows the resulting token-F1 score distributions against the VDC reference captions. The composed adapter remains close to the single-video adapter at both model scales. For FRAMES2LORA at 500M, the composed adapter retains 93.1% of the single-video adapter's mean token-F1, with a mean score of 0.206 compared to 0.221. At 2.2B, it retains 86.2%, with a mean score of 0.211 compared to 0.245.

6 Conclusion

We introduced **FRAMES2LORA**, showing that parametric video internalization is achievable: a Perceiver hypernetwork converts a video into a

LoRA adapter in a single forward pass, enabling a frozen VLM to answer queries with no visual tokens in context. Trained only on captioning, FRAMES2LORA is statistically non-inferior and equivalent to direct video-in-context inference across all five captioning benchmarks at both 500M and 2.2B scales, and transfers zero-shot to video QA on seven of eight benchmark-scale pairings. It remains stable at 1,024 frames where direct inference degenerates, achieves 6–76× lower query latency with up to 1,500× fewer answer-time tokens, and supports rank-space adapter composition for long-video internalization without dedicated training. Across token budgets, FRAMES2LORA uniquely combines near-constant query TTFT, scalable preprocessing costs below KV caching and token compression, and stable output quality at longer contexts.

7 Limitations

FRAMES2LORA demonstrates that video context can be internalized into generated adapter weights, enabling text-only querying after a one-time video processing step. Our current implementation trains a separate hypernetwork for each target VLM scale, and we evaluate it on the 500M and 2.2B SmolVLM2 backbones. Extending the same framework to additional VLM families, larger models, and shared or scale-transferable hypernetworks is an important direction for future work.

The present training setup uses captioning and summarization supervision. This makes transfer to video question answering a zero-shot setting, where answer style can differ from the direct video-in-context baseline. In particular, FRAMES2LORA sometimes produces more descriptive answers for short-answer QA, which can lower lexical-overlap metrics even when the answer is judged semantically appropriate. Future work can incorporate mixed captioning–QA supervision, answer-length control, or lightweight calibration for task-specific formats.

Because FRAMES2LORA converts a video into a compact adapter, the representation may emphasize high-level scene and event information over some fine-grained details. This is most relevant for tasks requiring precise camera, spatial, or object-level distinctions. Adaptive-rank adapters, richer internalization objectives, or more targeted supervision may improve preservation of these details.

Finally, our chunk-composition experiment is an initial two-chunk test. The result suggests that independently generated adapters can be combined in rank space, but the current operation does not explicitly model temporal order. More structured composition mechanisms and audio-visual internalization remain promising extensions.

8 Ethics Statement

Our research does not use any personally identifiable information (PII) and all datasets employed in this work are used in accordance with their respective licenses.

Acknowledgments

This research is partially supported by the NVIDIA Academic Grant Program.

References

Lucas Caccia, Alan Ansell, Edoardo M. Ponti, Ivan Vulic, and Alessandro Sordoni. 2025. Training plug-n-play knowledge modules with deep context distillation. *arXiv preprint arXiv:2503.08727*.

Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D. Manning. 2025. AuroraCap: Efficient, performant video detailed captioning and a new benchmark. In *International Conference on Learning Representations*.

Rujikorn Charakorn, Edoardo Cetin, Shinnosuke Uesaka, and Robert Lange. 2026. Doc-to-LoRA: Learning to instantly internalize contexts. *arXiv preprint arXiv:2602.15902*.

Xinlong Chen, Yuanxing Zhang, Chongling Rao, Yushuo Guan, Jiaheng Liu, Fuzheng Zhang, Chengru Song, Qiang Liu, Di Zhang, and Tieniu Tan. 2025a. VidCapBench: A comprehensive benchmark of video captioning for controllable text-to-video generation. In *Annual Meeting of the Association for Computational Linguistics*.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. 2025b. LongVILA: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*.

Jang Hyun Cho, Andrea Madotto, Effrosyni Mavroudi, Triantafyllos Afouras, Tushar Nagarajan, Muhammad Maaz, Yale Song, Tengyu Ma, Shuming Hu, Hanoona Rasheed, Peize Sun, Po-Yao Huang, Daniel Bolya, Suyog Jain, Miguel Martin, Huiyu Wang, Nikhila Ravi, Shashank Jain, Temmy Stark, and 10 others. 2025. PerceptionLM: Open-access data

and models for detailed visual understanding. *arXiv preprint*.

Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. 2024. Finevideo. https://huggingface.co/datasets/HuggingFaceFV/finevideo.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. 2025. Framefusion: Combining similarity and importance for video token reduction on large vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22654–22663.

David Ha, Andrew M. Dai, and Quoc V. Le. 2017. HyperNetworks. In *International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *IEEE International Conference on Computer Vision*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Conference on Empirical Methods in Natural Language Processing*.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. 2025. TokenPacker: Efficient visual projector for multimodal LLM. *International Journal of Computer Vision*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Annual Meeting of the Association for Computational Linguistics*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Shichen Lu, Tongtian Yue, Longteng Guo, Handong Li, Xingjian He, Si Liu, and Jing Liu. 2025. ViPE: Visual perception in parameter space for efficient video-language understanding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17764–17775.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. 2025. SmolVLM: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

Jesse Mu, Xiang Lisa Li, and Noah D. Goodman. 2023. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*.

Rui Qian, Xiao-Wen Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. 2024. Streaming long video understanding with large language models. In *Advances in Neural Information Processing Systems*.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2025. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *IEEE International Conference on Computer Vision*.

Haoning Wu, Dongxu Li, Bei Dong, Ziyang Lin, Ziyuan Li, Jiaming Bai, Jianing Han, Xiang Wang, Heung-Yeung Shum, Deli Zhao, and 1 others. 2024. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems*.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. NExT-QA: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yifan Xu, Xinhao Li, Yichun Yang, Desen Meng, Rui Huang, and Limin Wang. 2025. CaReBench: A fine-grained benchmark for video captioning and retrieval. *arXiv preprint arXiv:2501.00513*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. 2019. ActivityNet-QA: A dataset for understanding complex web videos via question answering. In *AAAI Conference on Artificial Intelligence*.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Conference on Empirical Methods in Natural Language Processing*.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024. Long context transfer from language to vision. *Transactions on Machine Learning Research*.

A LLM Judge Evaluation

We use an LLM judge for two purposes: reference-based quality scoring and reference-free output preservation. The judge is `Qwen/Qwen3-VL-30B-A3B-Thinking-FP8`, served locally with vLLM through an OpenAI-compatible API. For the main reported judge scores, we use text-only judging: the judge receives the task prompt, reference text, and model output, but no video frames. We set temperature to 0, use a maximum of 1024 output tokens for reference-based scoring, and request JSON-formatted outputs. For pure output similarity, we use the same judge with a maximum of 768 output tokens.

For reference-based quality, each candidate is scored independently against the reference. The judge is not shown model names. For auxiliary paired judgments, the direct baseline and FRAMES2LORA outputs are anonymized as Candidate A and Candidate B, and their order is randomized with a fixed seed. These paired judgments are used as an audit and are not the primary metric unless explicitly reported.

A.1 Reference-Based Captioning Judge

For captioning and description tasks, the judge measures semantic coverage of the reference caption. Extra details that are absent from the reference are not penalized unless they directly contradict the reference.

Captioning Judge Prompt

Task guidance: This is video caption/reference evaluation. Judge semantic coverage, factual precision, and task fit.

```
Prompt: {prompt}
Reference caption: {reference}
Candidate caption: {candidate}
```

Evaluate only semantic coverage of the reference caption for the requested caption task. Do not reward or penalize writing style, fluency, verbosity, or formatting, except when the candidate is invalid or impossible to understand. Extra details absent from the reference are not automatically wrong. Record notable extra details separately. Penalize only direct contradictions or missing reference facts. A direct contradiction requires that the candidate and reference cannot both be true.

Coverage scoring rubric: 5 = complete coverage; 4 = mostly complete; 3 = partial; 2 = weak overlap; 1 = minimal overlap; 0 = unrelated, invalid, nonsensical, or contradicts the main reference event.

Return only valid JSON with fields: `coverage_score` integer 0–5; `coverage_label`; `covered_reference_facts`; `missing_reference_facts`; `direct_contradictions`; `extra_details`; `extra_details_type`; `rationale`.

A.2 Reference-Based QA Judge

For QA tasks, the judge first extracts the answer implied by the model output and then compares it to the reference answer. This avoids over-penalizing verbose outputs that contain the correct answer.

QA Judge Prompt

Task guidance: This is video question answering. Judge semantic answer correctness. Allow paraphrases and indirect answers. For yes/no references, infer yes/no from the candidate if the candidate clearly implies it.

```
Question: {question}
Reference answer: {reference}
Candidate response: {candidate}
```

First extract the candidate's answer to the question, then compare that extracted answer to the reference. Do not require the candidate to be concise. A verbose response can be correct if it contains or clearly gives the answer. If the candidate describes the same scene but does not answer the requested attribute, action, location, or count, score it as low-to-partial rather than correct.

Scoring rubric: 5 = fully correct; 4 = correct main answer with minor missing specificity or harmless extra detail; 3 = partially correct; 2 = related but does not clearly answer; 1 = minimal overlap; 0 = contradiction, different answer, or invalid.

Return only valid JSON with fields: `extracted_answer`; `score` integer 0–5; `answer_label`; `contains_answer`; `direct_contradiction`; `extra_details_affect_score`; `missing_key_answer_parts`; `contradictory_parts`; `rationale`.

B Evaluation Prompts and Task Templates

This appendix provides the exact evaluation prompts and task-specific templates used across all the benchmarks in our experiments.

B.1 Video Captioning and Description Benchmarks

Below are the prompts used to generate descriptions for whole videos, clips, and specific features (e.g., spatial layout, temporal progression, and cinematography style).

ActivityNet Captions

"Describe what is happening in this video."

PLM-RDCap

"Describe what happens in this video."

PLM-RCap

"Describe what happens in this video clip."

Video Description Corpus (VDC)

Short Caption:
"Summarize this video in one detailed sentence."

Detailed Caption:
"Describe this video in detail."

Camera:
"Describe the camera work, framing, and viewpoint in this video."

Background:
"Describe the background, setting, and environment in this video."

Main Object:
"Describe the main subject and its actions in this video."

CaReBench

Caption:
"Describe the video in as much useful visual detail as possible. Include the main activity, visible people or objects, scene context, appearance, and any important visual details that help explain what is happening."

Events:
"Describe the key visible events in chronological order. Include all important actions and changes you can observe, with enough detail to distinguish each event clearly."

Objects:
"Describe the important visible objects and entities in the video in as much useful detail as possible. Include their appearance, location, and role in the scene when visible."

Spatial Caption:
"Describe the spatial layout in as much useful detail as possible: where the people, objects, and scene elements are located, how they are positioned relative to each other, and what parts of the scene are in the foreground, background, left, right, center, above, or below."

Temporal Caption: "Describe the temporal progression in as much useful detail as possible. Explain what happens over time, the order of visible actions, and how the scene or subjects change from the beginning to the end."

B.2 Video Question Answering (QA) Benchmarks

For question answering tasks, templates are structured to format the inputs and instructions depending on whether choices are provided (offered options) or hidden.

NExT-QA

{question}
Answer only the question, in one sentence.

ActivityNet-QA

{question}
Answer only the question, in one sentence.

PLM-SGQA

{question}
Answer only the question, in one sentence.

VidCapBench

{question}
Answer only the question, in one sentence.

C Rank-Direction Ablation

C.1 Setup

We test whether different rank directions in a generated LoRA adapter contribute unequally to cap-

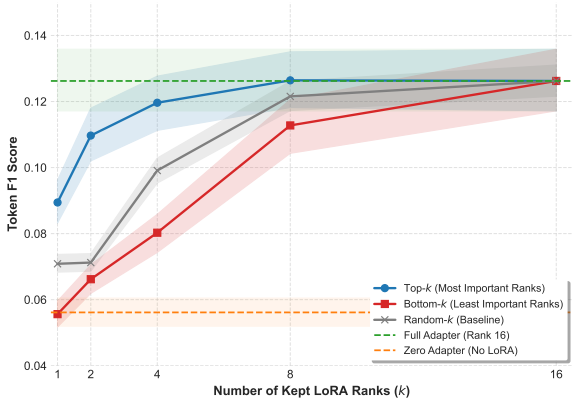


Figure 6: Rank-direction ablation on ActivityNet Captions. Top-k rank slices recover performance faster than random or bottom-k slices, suggesting that the Frobenius norm product is a useful heuristic for rank importance. The Remove-Top-k curve has a higher point estimate than the full adapter at small k, but this should be interpreted cautiously because confidence intervals overlap.

tioning performance. The ablation is run on 500 examples from the ActivityNet Captions evaluation split (Krishna et al., 2017), using the 2.2B FRAMES2LORA checkpoint. For each example, we generate the video-conditioned rank-16 adapter and decompose it into rank-slice pairs $\{(A_r, B_r)\}_{r=1}^{16}$, where $A_r \in \mathbb{R}^{1\times d_{\text{in}}}$ and $B_r \in \mathbb{R}^{1\times d_{\text{out}}}$. Under our row-vector implementation, rank slice r contributes

$$\Delta y_r = s\,(xA_r^\top)B_r.$$

We score each slice by the Frobenius norm product

$$\|A_r\|_F \cdot \|B_r\|_F.$$

We evaluate four selection strategies across budgets $k \in \{1, 2, 4, 8, 16\}$:

- **Top-k:** retain the k highest-scoring rank slices.
- **Bottom-k:** retain the k lowest-scoring rank slices.
- **Random-k:** retain k randomly selected slices, averaged over 3 seeds.
- **Remove-Top-k:** remove the k highest-scoring slices and retain the remaining $16 - k$.

We report Token-F1 against reference captions with 95% bootstrap confidence intervals over examples.

C.2 Analysis

Table 5 reports the numerical ablation results, and Figure 6 visualizes the same rank-pruning trajectories.

Rank directions are redundant but not exchangeable. The generated adapters are compressible along the rank dimension. Retaining the top-8 rank slices gives a Token-F1 of 0.1264, close to the full rank-16 adapter score of 0.1262. At $k = 4$, the top-k adapter reaches 0.1196, which is 94.8% of the full adapter's absolute Token-F1 and recovers 90.6% of the improvement over the zero-adapter baseline. This suggests that much of the useful adaptation is concentrated in a subset of rank directions.

Norm product is a useful heuristic for rank importance. The Frobenius norm product separates useful from less useful directions. At $k = 1$, Top-k reaches 0.0894, while Bottom-k reaches 0.0556, slightly below the zero-adapter baseline of 0.0561. Random-k generally falls between Top-k and Bottom-k at matched budgets. Thus, high-norm rank slices tend to be more useful, although the norm product should be treated as a heuristic rather than a complete causal explanation.

Removing the dominant direction has a higher point estimate. Removing the highest-norm rank slice gives a higher point estimate than the full adapter, increasing Token-F1 from 0.1262 to 0.1317. Removing the top four slices also remains close to the full adapter at 0.1275. Since the confidence intervals overlap, we treat this as suggestive rather than conclusive. One possible explanation is that the dominant direction captures a generic captioning prior, and removing it shifts generation toward more video-specific directions.

Rank ordering is stable across examples. The rank ordering is highly consistent across the 500 examples: rank direction R11 is the highest-scoring direction in all examples, while R7 is consistently among the lowest-scoring directions. This suggests that the hypernetwork learns a stable output coordinate system for rank directions, rather than assigning importance arbitrarily for each video.

k	Top-k	Bottom-k	Random-k	Remove-Top-k
0 (Zero)	0.0561 [.052,.060]	0.0561	0.0561	0.0561
1	0.0894 [.083,.096]	0.0556 [.052,.060]	0.0709 [.068,.074]	**0.1317** [.123,.141]
2	0.1097 [.102,.118]	0.0662 [.062,.071]	0.0712 [.069,.074]	0.1277 [.118,.137]
4	0.1196 [.111,.128]	0.0803 [.074,.086]	0.0991 [.095,.103]	0.1275 [.119,.137]
8	0.1264 [.118,.135]	0.1128 [.104,.121]	0.1215 [.117,.126]	0.1128 [.104,.121]
16 (Full)	0.1262 [.117,.136]	0.1262	0.1262	0.0561

Table 5: Token F1 scores under rank-direction ablation on ActivityNet Captions. Brackets denote 95% confidence intervals. *Full Adapter* ($k = 16$) and *Zero Adapter* ($k = 0$) serve as upper and lower baselines.

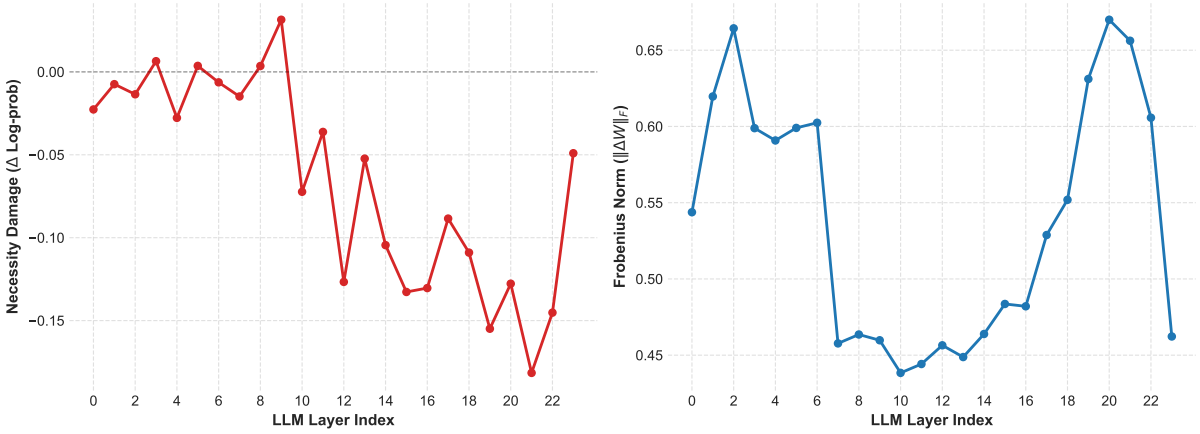


Figure 7: Layer-wise adapter-removal diagnostic. **Left:** signed removal effect from zeroing one layer's generated LoRA update; negative values indicate that removing the layer lowers the score. **Right:** Frobenius norm $\|\Delta W\|_F$ of generated LoRA weights across layers.

D Interpreting Hypernetwork-Generated Adapters

D.1 Setup

We use two diagnostic interventions to study how generated adapters affect the frozen 2.2B answer model: layer-wise adapter removal and direct logit attribution. The experiments are run on CareBench diagnostic examples, including caption and spatial-caption prompts.

Each example is scored by teacher-forced log-probability under the frozen answer model with the generated adapter active. Since these diagnostics use open-ended reference strings, we score each reference string and use the highest-scoring reference for the diagnostic. Candidate strings may contain multiple tokens, so we score a candidate string z by length-normalized teacher-forced log-probability:

$$\ell(z \mid p) = \frac{1}{|z|} \sum_{t=1}^{|z|} \log P(z_t \mid z_{<t}, p). \tag{10}$$

The scalar diagnostic score is therefore

$$\mathcal{S} = \max_{r \in \mathcal{R}} \ell(r \mid p), \tag{11}$$

where $\mathcal{R}$ is the set of reference strings for the example.

For direct logit attribution, we need a direction in the output-embedding space. We use the mean output embedding of the selected reference tokens and denote the normalized direction by $\hat{d}$. This gives a single diagnostic direction toward the reference answer/caption.

D.2 Layer-Wise Adapter Removal

For each transformer layer ℓ, we zero out only the generated LoRA update at that layer and recompute the diagnostic score. We report the signed removal effect

$$\text{Effect}_\ell = \mathcal{S}_{\text{without } \ell} - \mathcal{S}_{\text{full}}. \tag{12}$$

Negative values indicate that removing the layer lowers the score, so the layer's adapter update is useful under this diagnostic. Values near zero

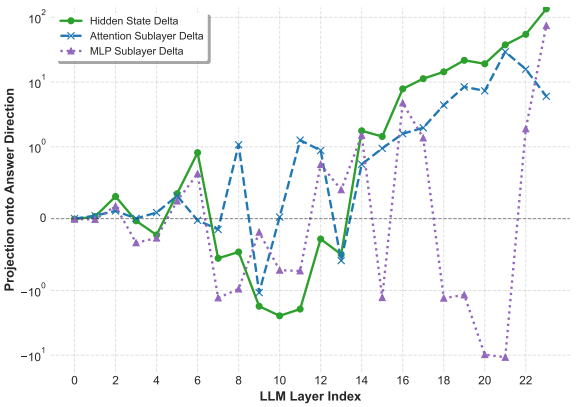


Figure 8: Direct logit attribution of adapter-induced representation shifts projected onto the diagnostic answer direction across 24 LLM layers. Later layers show the largest alignment with the answer direction, suggesting late-layer logit steering.

Setting	500M	2.2B
Training steps	9,000	7,000
GPUs	4×A100	6×A100
Wall-clock training time	37 hours	201 hours
Per-device batch size	48	8
Gradient accumulation steps	2	5
Effective batch size	384	240
LoRA rank	16	16
Sampled frames	12	12
Max video dimension	384 px	384 px
Perceiver latent size	512	512
Learning rate	1×10^{-4}	1×10^{-4}
Warmup ratio	0.03	0.03
Weight decay	0.01	0.01

Table 6: Training configuration for the 500M and 2.2B FRAMES2LORA runs. Wall-clock training time reports elapsed training time, not total GPU-hours. Effective batch size is computed as number of GPUs × per-device batch size × gradient accumulation steps.

indicate little measurable effect from removing that layer.

Figure 7 shows a mismatch between generated-weight norm and functional effect. Some early layers receive relatively large LoRA updates, but removing them changes the diagnostic score only weakly. In contrast, several later layers produce larger negative removal effects, indicating that their adapter updates matter more for the scored prediction. This suggests that the adapter is not used uniformly across the transformer stack: early updates may shape intermediate representations, while later updates appear more directly connected to the final answer/caption likelihood.

This also shows that Frobenius norm alone is not a complete measure of adapter importance. Large generated weights can be weakly causal under this intervention, whereas smaller or comparable later-layer updates can have stronger effects on the output score. We therefore interpret the result as a norm–function dissociation, not as a full causal explanation of the adapter mechanism.

D.3 Direct Logit Attribution

We next ask where the adapter-induced representation shift becomes aligned with the diagnostic target direction. Let

$$\Delta x_\ell = x_\ell^{\text{adapter}} - x_\ell^{\text{base}}$$

denote the residual-stream shift at layer ℓ, and let Δa_ℓ and Δm_ℓ denote the corresponding attention and MLP sublayer shifts. We project these shifts onto the diagnostic answer direction:

$$\begin{aligned} \mathrm{DLA}_\ell &= \Delta x_\ell \cdot \hat{d}, \\ \mathrm{DLA}_\ell^{\text{attn}} &= \Delta a_\ell \cdot \hat{d}, \\ \mathrm{DLA}_\ell^{\text{MLP}} &= \Delta m_\ell \cdot \hat{d}. \end{aligned} \tag{13}$$

Figure 8 shows that the adapter-induced shift is weakly aligned with the diagnostic direction in early and middle layers, but becomes much more aligned in later layers. This matches the layer-removal result: the adapter's effect becomes most visible close to the output logits.

The sublayer breakdown suggests that both attention and MLP components contribute to this late-stage steering. Rather than claiming that the generated adapter implements a specific memory mechanism, we interpret the pattern more conservatively: FRAMES2LORA appears to induce representation changes throughout the network, but the changes most directly aligned with the target answer/caption emerge in later layers.

E Training Details

Table 6 summarizes the main training configuration for the two FRAMES2LORA model scales. In both runs, only the hypernetwork parameters are trained; the video encoder and answer model remain frozen.

Both models use rank-16 generated LoRA adapters, 12 uniformly sampled frames, a maximum video dimension of 384 pixels, Perceiver latent size 512, learning rate 1×10^{-4}, warmup ratio

0.03, and weight decay 0.01. The 500M model is trained for 9,000 steps on 4 A100 GPUs for 37 wall-clock hours, with per-device batch size 48 and gradient accumulation 2, giving an effective batch size of 384. The 2.2B model is trained for 7,000 steps on 6 A100 GPUs for 201 wall-clock hours, with per-device batch size 8 and gradient accumulation 5, giving an effective batch size of 240.

F Additional Results

This section provides additional experimental evaluations on video-grounded multimodal judging and long-form video understanding.

F.1 Video-Grounded VLM Judge Evaluation

We employ `Qwen3-VL-30B-A3B-Thinking-FP8` with sampled video frames passed directly as visual input alongside the task prompt and reference. The evaluation prompt instructs the multimodal judge to verify visual grounding directly against the video events, explicitly penalizing hallucinations, unsupported visual details, or factual contradictions with respect to the video content. Scores are evaluated on the 1–5 scale and rescaled to $[0, 1]$.

Results and Analysis. Table 7 reports the video-grounded judge scores across all 9 captioning and question answering benchmarks for both SmolVLM 500M and 2.2B scales.

The video-grounded results corroborate the findings from the text-only evaluation:

- **Directional Consistency:** Across all 5 captioning benchmarks and 4 QA benchmarks, the relative preservation patterns closely mirror the main text-only judge. FRAMES2LORA maintains strong visual grounding across tasks.
- **Question Answering Parity:** On QA benchmarks, FRAMES2LORA achieves near-parity or surpasses the base model. Notably, on SmolVLM 500M, FRAMES2LORA outperforms the base model on NExT-QA ($+0.024$) and ActivityNet-QA ($+0.052$), demonstrating that causal, temporal, and fine-grained visual information is faithfully retained in the parameter space under direct video inspection.

F.2 Long-Video Evaluation on LongVideoBench

Motivation and Setup. To evaluate scalability on long-form video understanding beyond short-to-medium clips, we evaluate on LONGVIDEOBENCH (Wu et al., 2024). LONGVIDEOBENCH consists of diverse, long-context interleaved video tasks with video durations averaging 4–8 minutes.

We evaluate both the base video-in-context model and FRAMES2LORA under varying uniformly sampled frame counts $F \in \{12, 48, 128\}$ on both SmolVLM 500M and 2.2B backbones. Candidate generations are evaluated using our LLM judge on the standard 1–5 scale.

Results and Analysis. As shown in Table 8, FRAMES2LORA maintains robust performance across frame counts on long videos. At 128 frames on the 2.2B scale, FRAMES2LORA outperforms direct video-in-context inference (1.86 vs. 1.49). In high-frame, long-video regimes, direct video-in-context inference suffers from context-window degradation and visual token crowding, whereas FRAMES2LORA internalizes the visual representation into model weights, preserving coherent and video-faithful generation without context saturation.

F.3 Distribution Plots

Figures 9 and 10 show the LLM-judge score distributions and per-example score differences. Figures 11 and 12 show the corresponding token-F1 distributions and differences.

F.4 Spider Plots

Figures 13 and 14 show the QA and captioning spider plots.

G Qualitative Examples

Qualitative examples are shown in Figure 15, Figure 16, Figure 17, Figure 18, Figure 19, Figure 20, Figure 21, Figure 23, Figure 24, Figure 25, Figure 26, Figure 27, Figure 28, Figure 29, Figure 30, Figure 22.

H Use of AI Assistants

AI assistants were employed during this research to assist with writing refinement, grammar and paraphrasing checks, LaTeX formatting, and code implementation. All conceptual developments, experimental designs, empirical analyses, and final text were independently verified, reviewed, and finalized by the authors, who assume full responsibility for all content presented in this paper.

Benchmark	SmolVLM 500M			SmolVLM 2.2B		
	Base	V2L	Δ	Base	V2L	Δ
Video Captioning						
ActivityNet Captions	0.678	0.572	-0.106	0.744	0.682	-0.062
PLM-RDCap	0.530	0.433	-0.097	0.546	0.548	+0.002
PLM-RCap	0.358	0.322	-0.036	0.386	0.372	-0.014
VDC (aggregate)	0.522	0.444	-0.078	0.586	0.488	-0.098
CaReBench	0.390	0.314	-0.076	0.532	0.416	-0.116
Average	0.496	0.417	-0.079	0.559	0.501	-0.058
Video Question Answering						
NExT-QA (open)	0.534	0.558	+0.024	0.658	0.632	-0.026
ActivityNet-QA	0.626	0.678	+0.052	0.770	0.665	-0.105
PLM-SGQA	0.460	0.438	-0.022	0.552	0.530	-0.022
VidCapBench	0.558	0.456	-0.102	0.614	0.528	-0.086
Average	0.545	0.533	-0.012	0.649	0.589	-0.060

Table 7: Video-grounded evaluation using a multimodal VLM judge (`Qwen3-VL-30B-A3B-Thinking-FP8` with actual video frames provided as input) across all 9 captioning and QA benchmarks. Scores are rescaled to $[0, 1]$.

Model Scale	Frames	Base	FRAMES2LORA	Δ
SmolVLM 500M	12f	1.50	1.77	+0.27
	48f	1.43	1.75	+0.32
	128f	1.94	1.64	-0.30
SmolVLM 2.2B	12f	1.72	1.66	-0.06
	48f	1.83	1.67	-0.16
	128f	1.49	**1.86**	**+0.37**

Table 8: Evaluation on LONGVIDEOBENCH (videos averaging 4–8 minutes) across frame counts using LLM judge scores (1–5 scale). FRAMES2LORA scales effectively with frame count and outperforms the base model at 128 frames (2.2B).

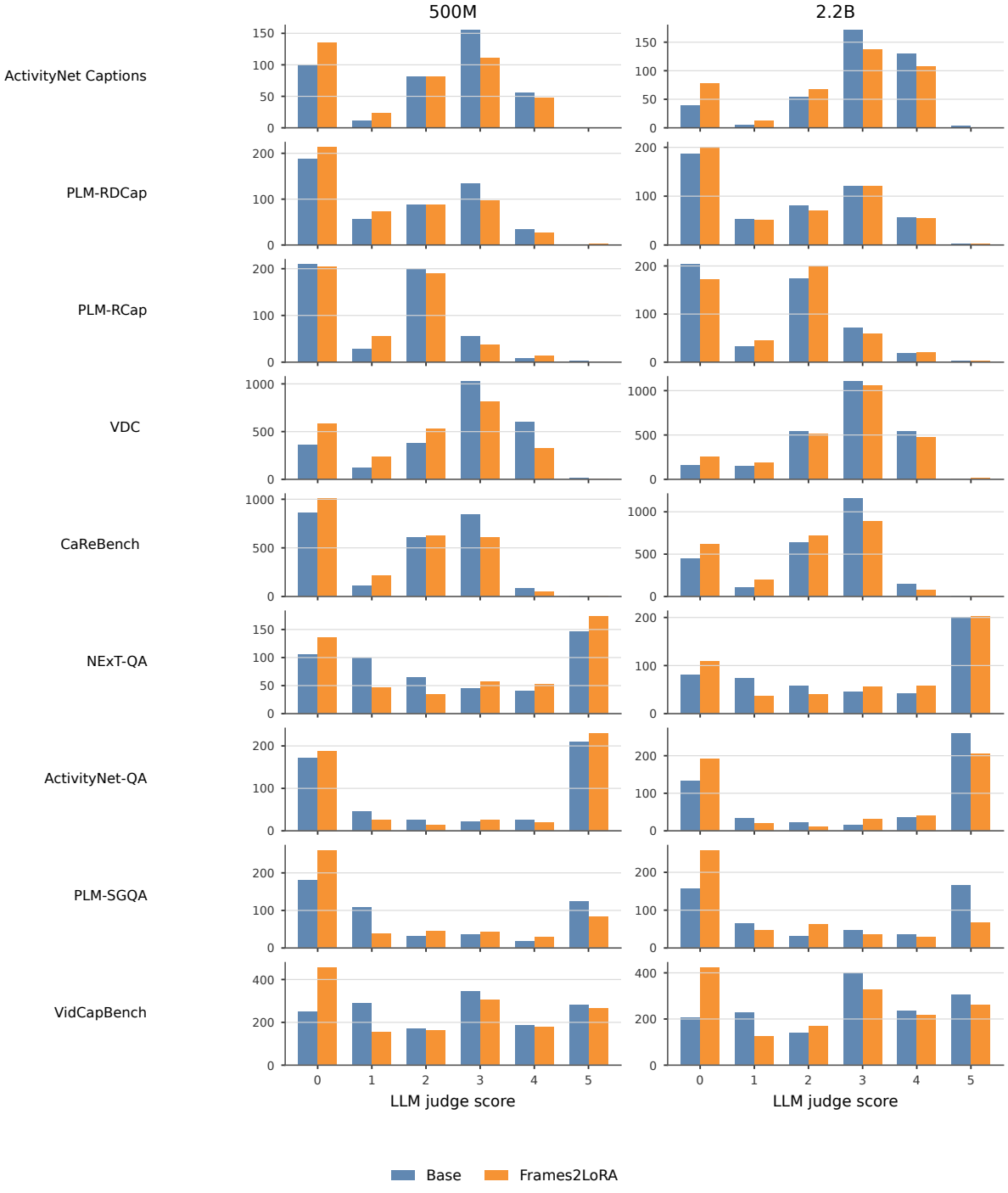


Figure 9: LLM-judge score distributions for the direct baseline and FRAMES2LORA.

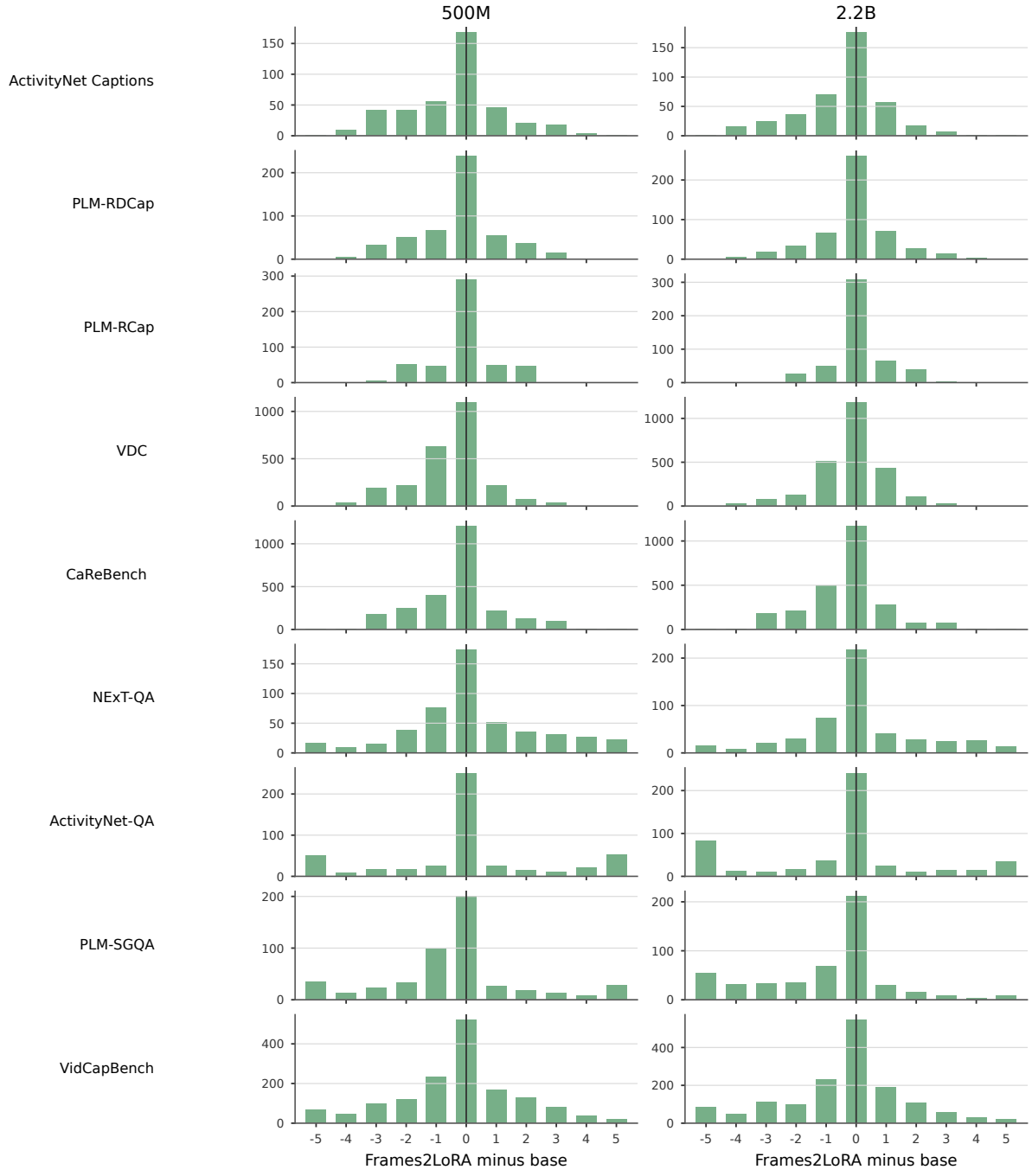


Figure 10: Per-example LLM-judge score differences between FRAMES2LORA and the direct baseline.

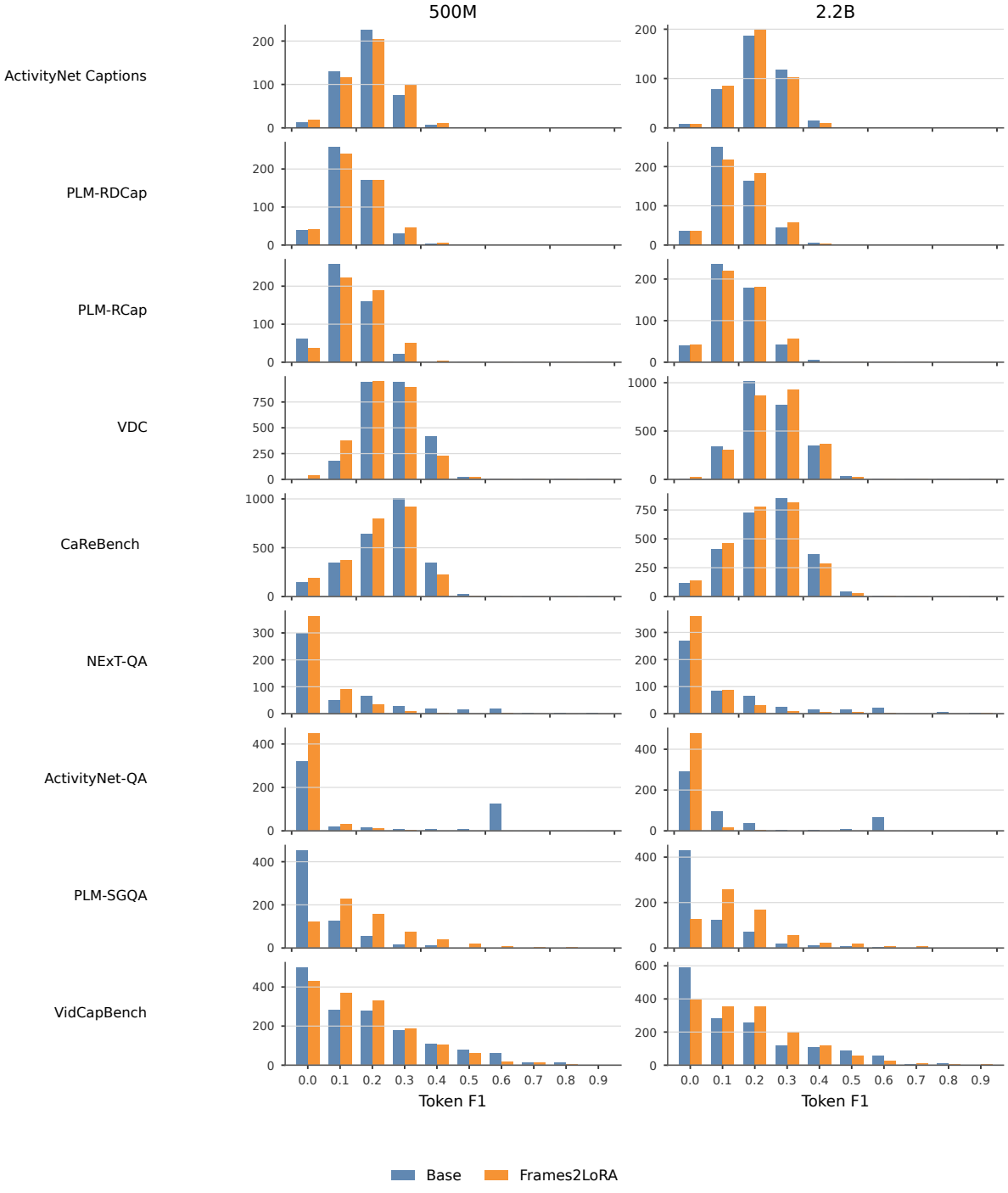


Figure 11: Token-F1 distributions for the direct baseline and FRAMES2LORA.

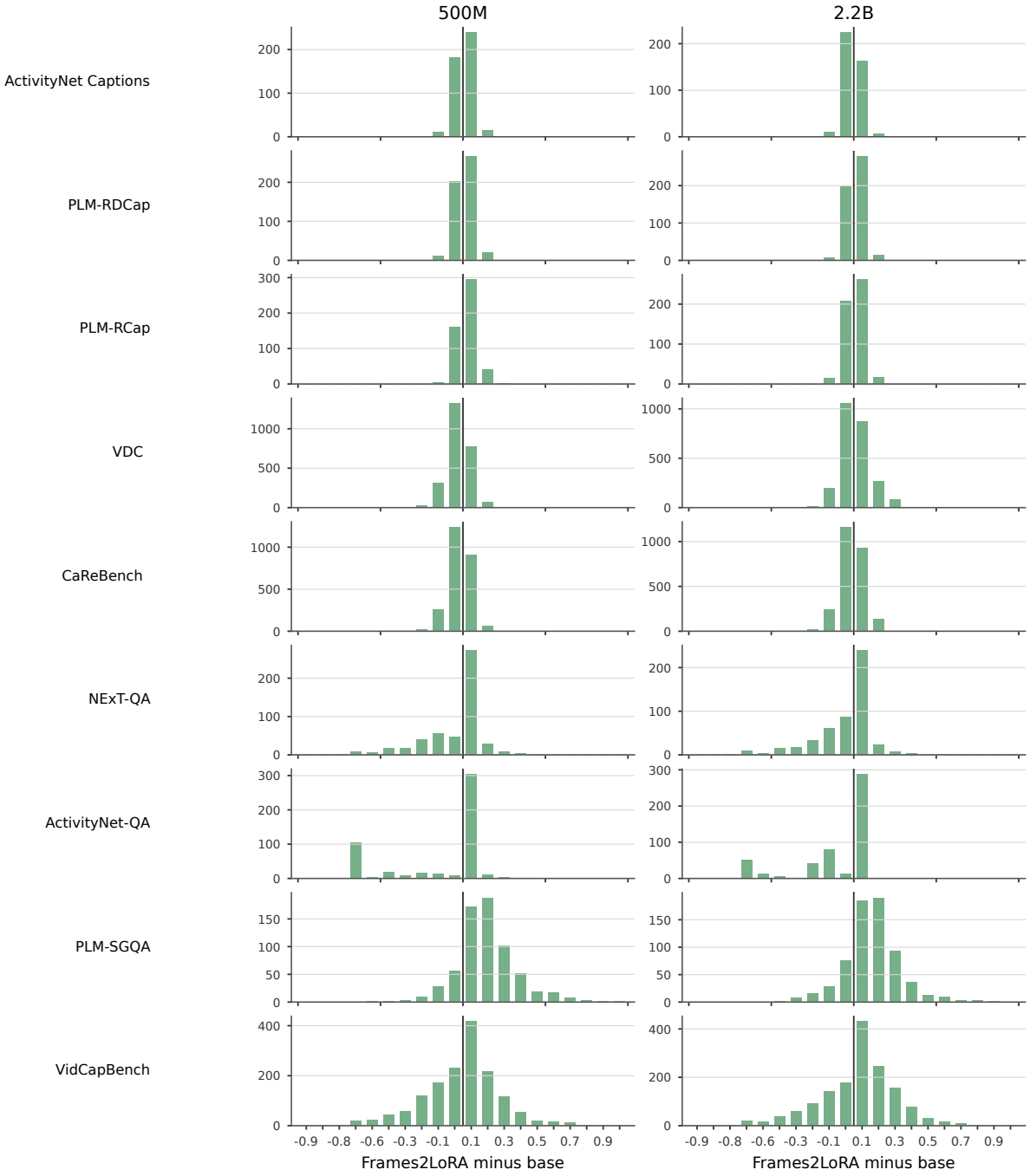


Figure 12: Per-example token-F1 differences between FRAMES2LORA and the direct baseline.

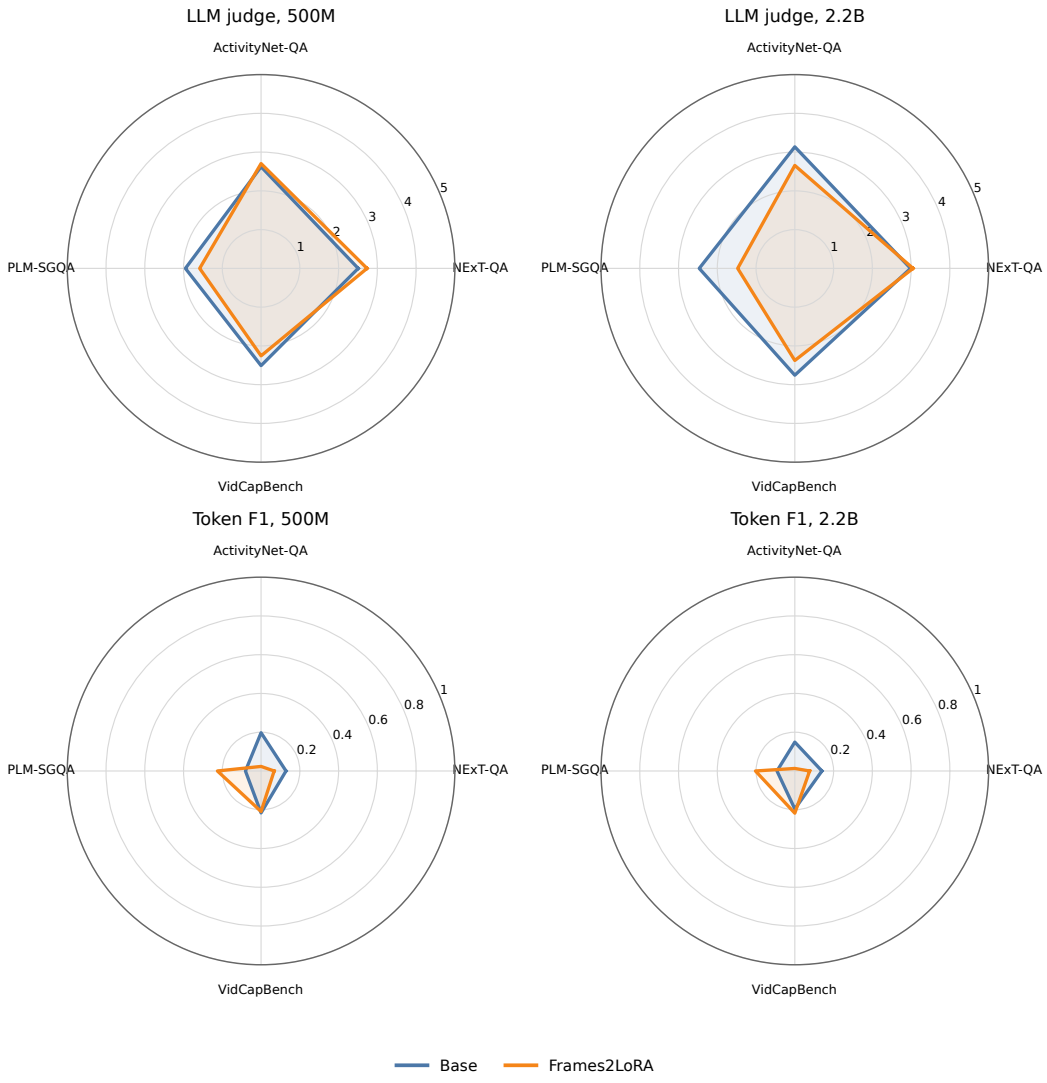


Figure 13: Spider plot for video question answering benchmarks.

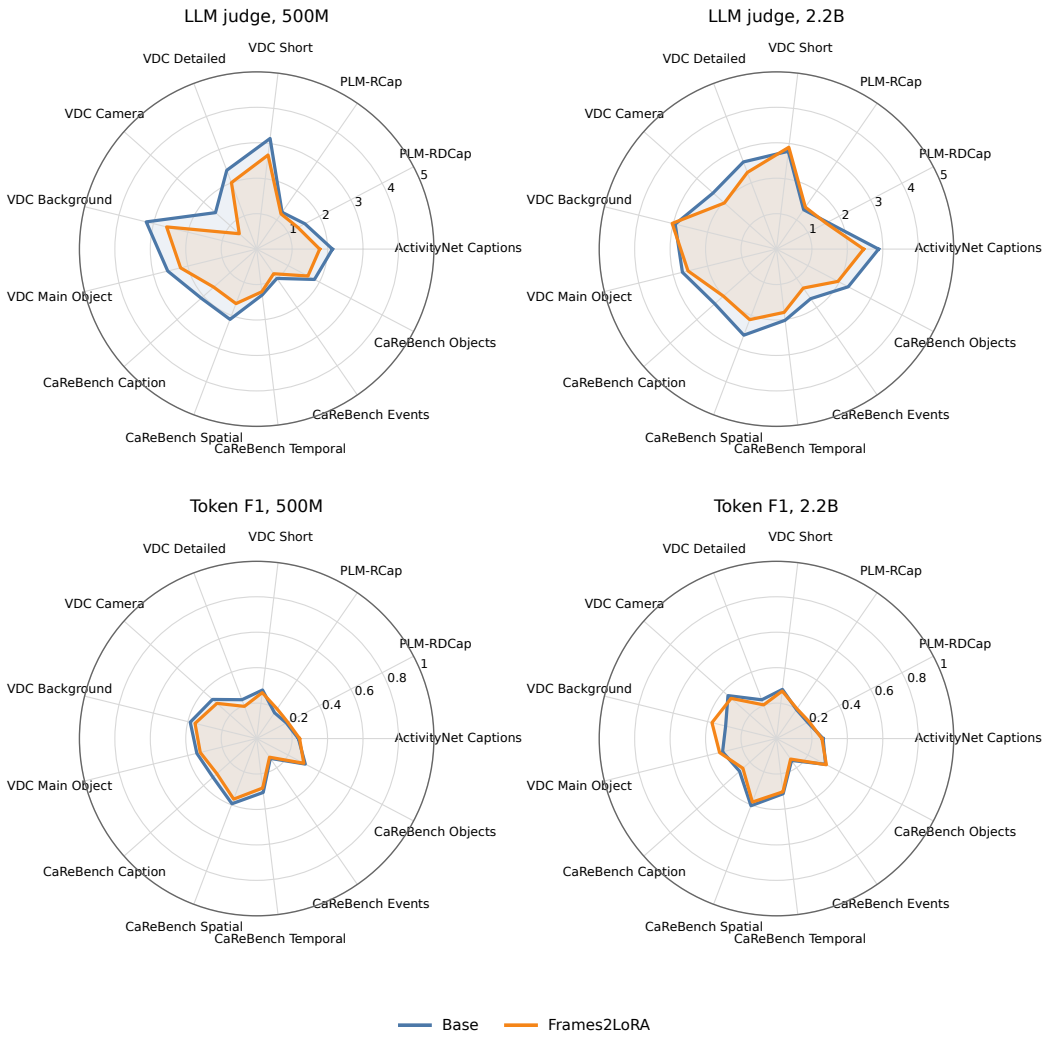


Figure 14: Spider plot for video captioning benchmarks.

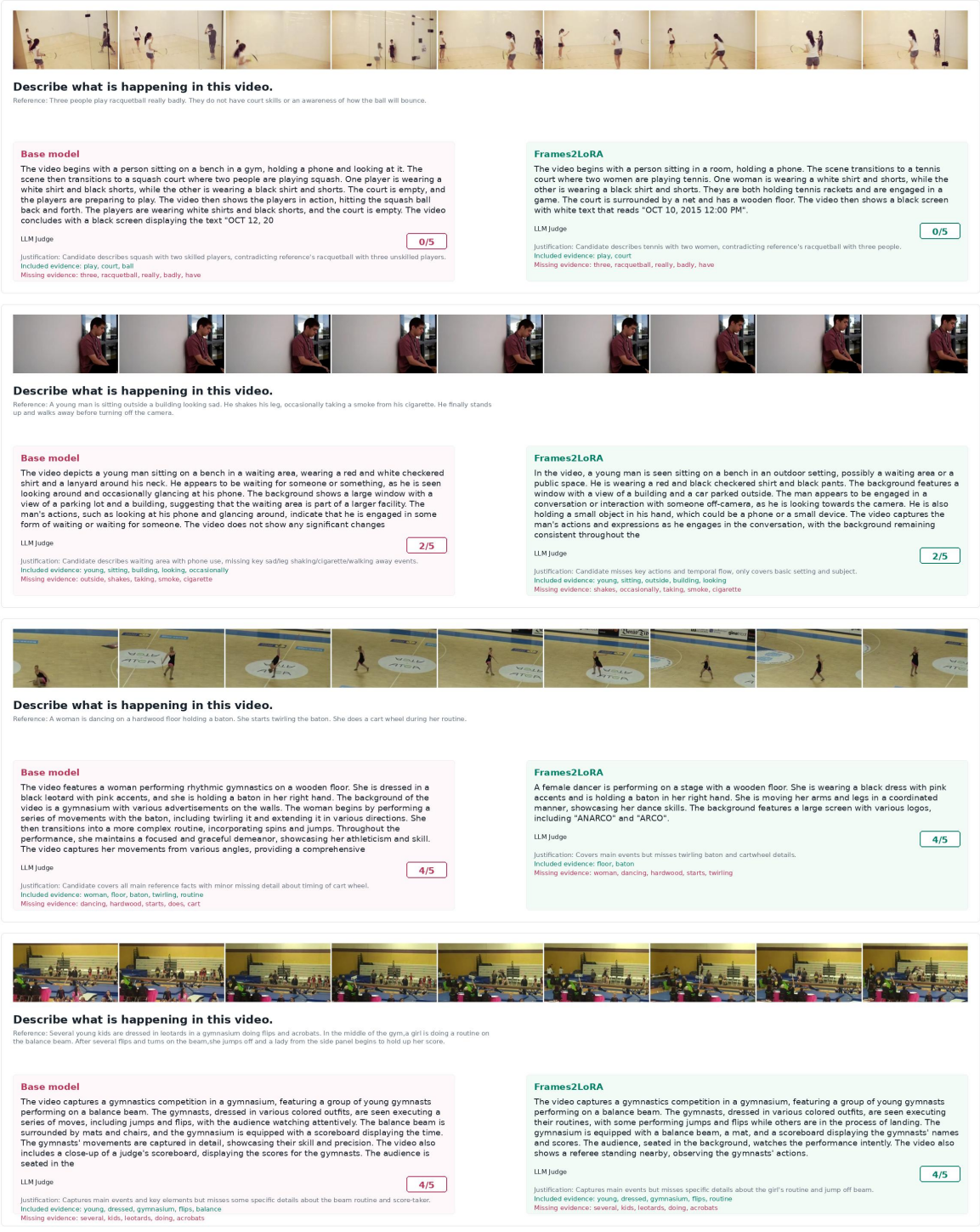


Figure 15: Qualitative examples from ActivityNet Captions.

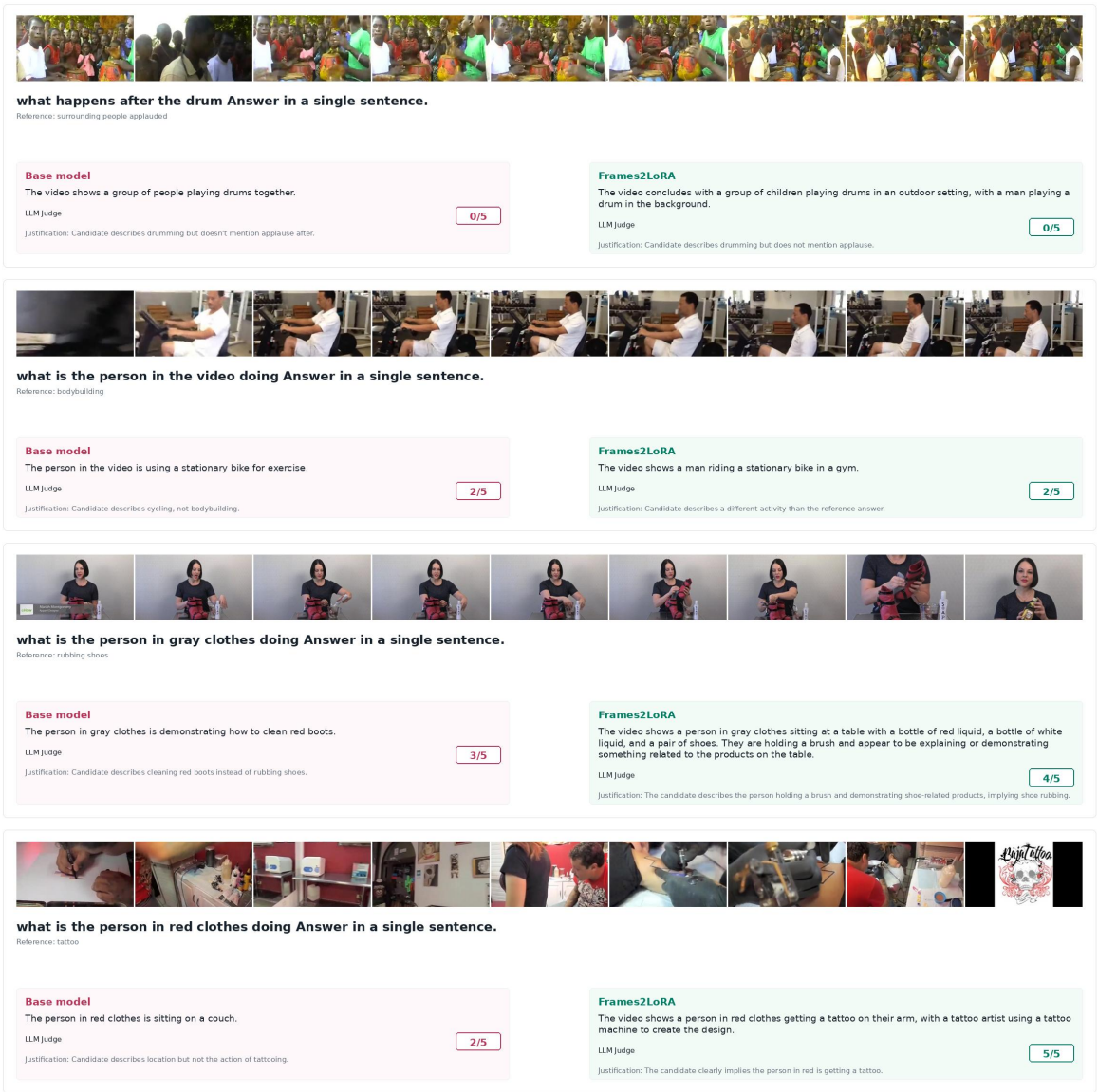
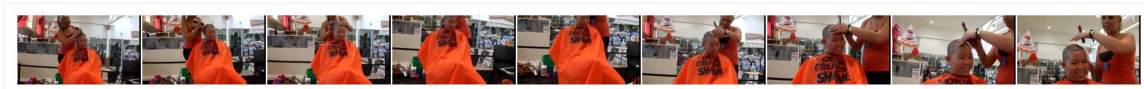


Figure 16: Qualitative examples from ActivityNetQA.



Describe the video in as much useful visual detail as possible. Include the main activity, visible people or objects, scene context, appearance, and any important visual details that help explain what is happening.

Reference: This video captures a woman giving a haircut to a child. The child is seated in a barber chair, draped in a bright orange cloth to prevent hair from falling onto his clothes, which feature black lettering across the chest area. The woman, standing behind him, is dressed in a rose-red sleeveless top and black shorts, with her blonde hair tied up in a high ponytail. In her right hand, she holds a black electric hair clipper. They are inside a barbershop, facing the camera. In the video, the woman begins cutting hair from the front right side of the child's head, her movements swift and precise as she works her way to the back. The camera then zooms in, showing her starting to trim the left side of his hair. Finally, she uses her right hand to tousle the hair at the front of the child's head to signify that the haircut is complete. The footage is shot from a third-person perspective, effectively showcasing the woman's rapid haircutting technique.

Base model

A woman is cutting a boy's hair in a salon. The boy is sitting in a chair with an orange cape on. The woman is using a pair of scissors to cut the boy's hair.

LLM Judge 2/5

Justification: Candidate misses most reference facts and contradicts key details like using scissors instead of clipper.
Included evidence: woman, chair, orange, hair
Missing evidence: captures, giving, haircut, child, seated

Frames2LoRA

A woman is getting her hair cut in a salon. She is wearing an orange cape and is sitting in a chair. The stylist is cutting her hair with a pair of scissors.

LLM Judge 1/5

Justification: Candidate misidentifies subject and tool, missing all key reference details.
Included evidence: woman, chair, orange, hair
Missing evidence: captures, giving, haircut, child, seated



Describe the video in as much useful visual detail as possible. Include the main activity, visible people or objects, scene context, appearance, and any important visual details that help explain what is happening.

Reference: The video features a man shown in full face as he makes a phone call. Before the call connects, two lines of subtitles appear at the bottom of the screen, reading "(618) 465-4545 if you ever need the time and temperature in Alton, Illinois." The man picks up a wired phone that is designed with a strong glossy finish and multiple facets in his right hand. He gazes into the camera and speaks into the microphone, then he purses his lips, blinks a few times, raises an eyebrow, and continues to speak. Afterward, he hangs up the phone. The video is shot from a first-person perspective, primarily aiming to promote and disseminate the phone number mentioned in the subtitles. It provides a detailed demonstration of the phone dialing service and its offerings through the man's actual phone call experience. Although the video is quite short, it effectively illustrates the convenience and practicality of calling to inquire about the time and weather.

Base model

The video shows a person holding a phone to their ear. The person is wearing a white shirt and has short hair. The background is a room with a white ceiling and a poster on the wall. The person is talking on the phone and the text on the screen reads "(618) 465-4545 if you ever need the time and temperature in Alton, Illinois."

LLM Judge 3/5

Justification: Candidate covers basic phone call and text but misses key details about the man, phone, and purpose.
Included evidence: phone, screen, 4545, ever
Missing evidence: shown, full, face, makes, call

Frames2LoRA

A man is holding a phone to his ear. He is wearing a black shirt and has a beard. The background is a room with a poster on the wall.

LLM Judge 2/5

Justification: Candidate misses key reference facts about the call sequence and promotional purpose
Included evidence: phone
Missing evidence: shown, full, face, makes, call



Describe the video in as much useful visual detail as possible. Include the main activity, visible people or objects, scene context, appearance, and any important visual details that help explain what is happening.

Reference: This video captures a little boy as he ties his own shoelaces. The boy is dressed in a red and blue plaid shirt paired with red shorts. He is sitting on a staircase, which has wooden steps and white railings. The boy has a blue sneaker on his left foot, while the right shoe rests beside him. In the video, the boy lowers his head to focus on tying his shoelaces. He grips both ends of the lace, trying to wrap them around each other to make a knot. However, one of the laces falls to the ground. Afterward, he successfully ties both ends of the lace together and pulls tightly on them, securing the knot. At this point, the camera zooms in for a close-up of the boy's feet. Next, he forms a loop with the left lace and tucks the left lace through the loop. Finally, he tightens both ends, resulting in a neatly tied bow. The video is shot from a third-person perspective, showcasing the boy's still-developing skill in shoelace tying, but his patience is commendable.

Base model

A young boy is sitting on a wooden staircase, tying his blue sneakers. He is wearing a plaid shirt and red shorts. The boy is focused on tying the laces of his shoes.

LLM Judge 4/5

Justification: Captures main activity and setting but misses key details of shoelace tying process
Included evidence: blue, plaid, shirt, shorts, sitting
Missing evidence: captures, little, ties, shoelaces, dressed

Frames2LoRA

A young boy is sitting on a wooden staircase, wearing a plaid shirt and blue shorts. He is tying his shoelaces.

LLM Judge 4/5

Justification: Captures main action and setting but misses key details about clothing, shoe position, and lacing steps.
Included evidence: shoelaces, blue, plaid, shirt, shorts
Missing evidence: captures, little, ties, dressed, paired



Describe the video in as much useful visual detail as possible. Include the main activity, visible people or objects, scene context, appearance, and any important visual details that help explain what is happening.

Reference: This video depicts an elderly man meticulously cleaning water stains off a glass window using a squeegee. He is dressed in a gray-blue plaid shirt, and his hair has turned mostly white, suggesting he is quite advanced in age. In his right hand, he holds the squeegee, while in front of him stands a window with a black frame. Inside the room, there are several wooden desks, their surfaces a warm yellow hue, each equipped with a drawer. A whiteboard for writing hangs on the wall, and a water dispenser sits in the corner of the room. The elderly man has a red cord around his neck, from which hangs a black card holder.In the video, the gentleman is positioned outside the building, skillfully scraping from the upper left corner of the window toward the right. He moves down to the bottom, then creates a sweeping arc back to the lower left corner, followed by another arc over to the right, repeating this motion until all the water stains are thoroughly removed. The footage is captured from the right side of the elderly man as he demonstrates the technique for cleaning the window, explaining the process aloud, likely teaching the videographer how to effectively wipe away the water stains.

Base model

A man wearing a checkered shirt and a red lanyard is cleaning a glass window in a classroom. He is using a squeegee to clean the window, which is covered in streaks of dirt and grime. The classroom has white walls and a black floor.

LLM Judge 3/5

Justification: Candidate covers basic action but misses key details and has minor contradiction.
Included evidence: cleaning, glass, window, using, squeegee
Missing evidence: depicts, elderly, meticulously, water, stains

Frames2LoRA

A man wearing a checkered shirt and a lanyard is cleaning a glass door with a squeegee. He is using a long-handled tool to clean the glass, starting from the top and moving down. The glass door is framed by a black frame and has a white wall behind it. There are desks and chairs visible in the background, suggesting an office or classroom setting. The man is focused on his task, ensuring the glass is clean.

LLM Judge 4/5

Justification: Covers main action and setting but misses key details about man's appearance and room features.
Included evidence: cleaning, glass, using, squeegee, shirt
Missing evidence: depicts, elderly, meticulously, water, stains

Figure 17: Qualitative examples from CaReBench: Caption.

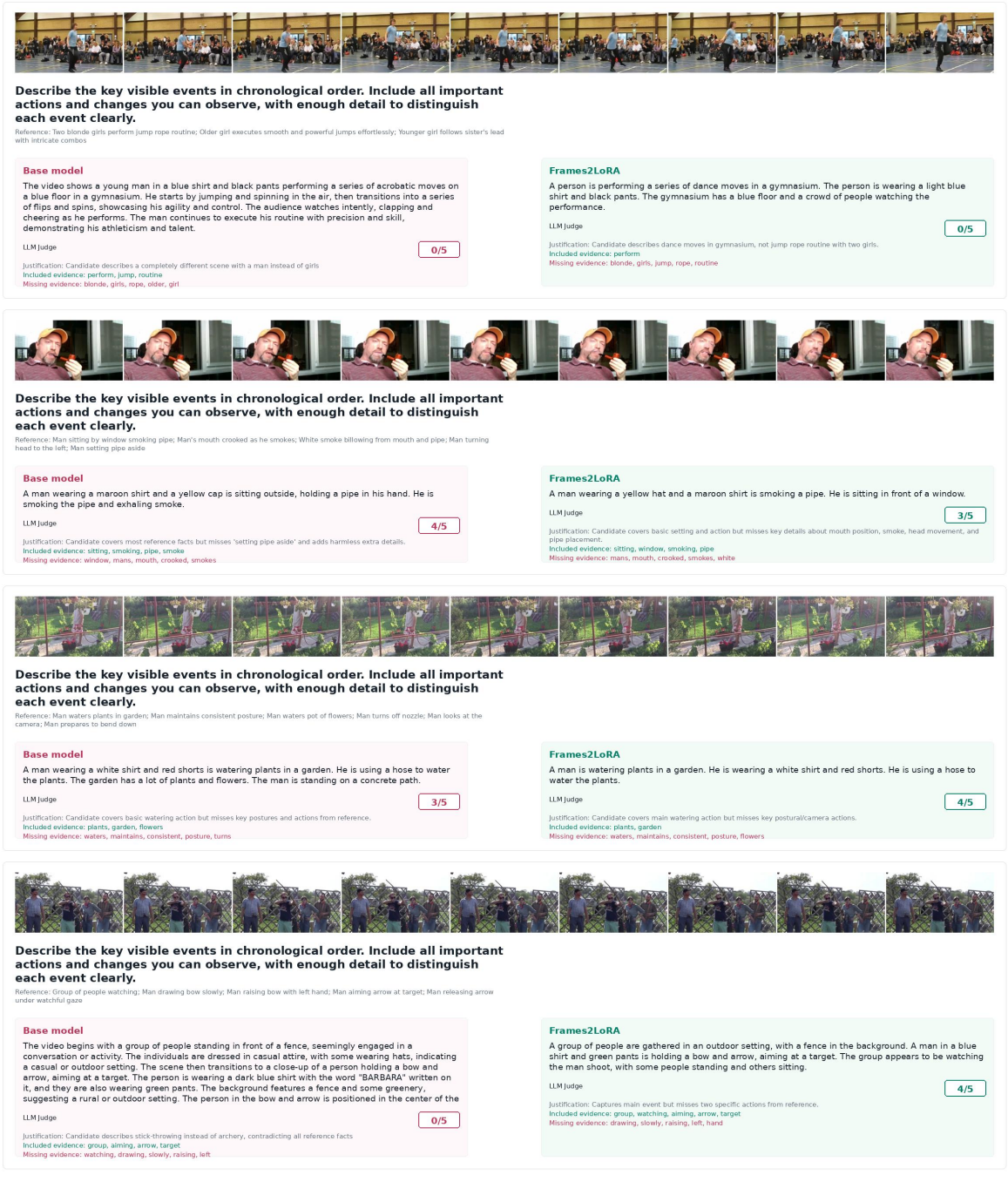


Figure 18: Qualitative examples from CaReBench: Events.

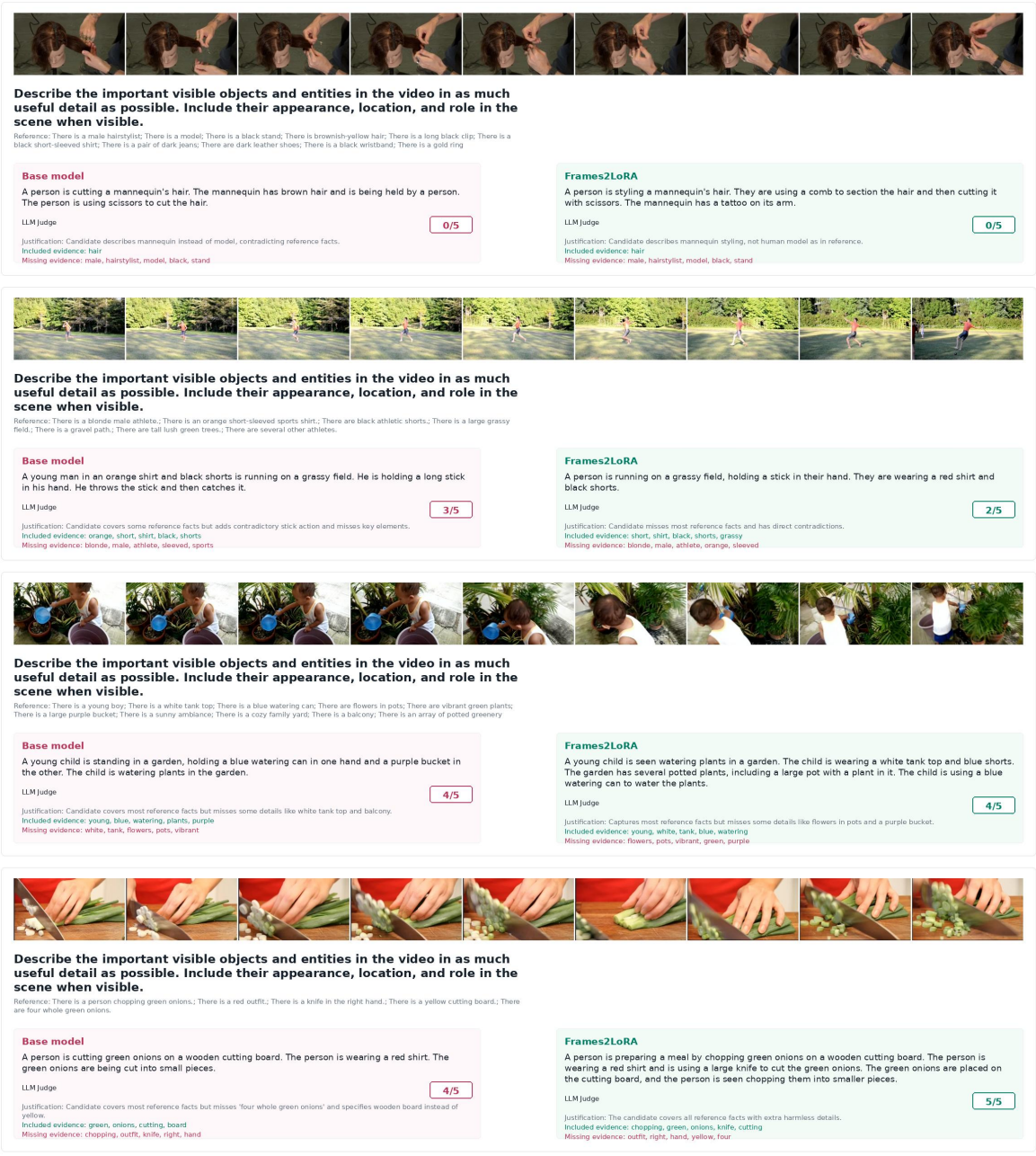


Figure 19: Qualitative examples from CaReBench: Objects.

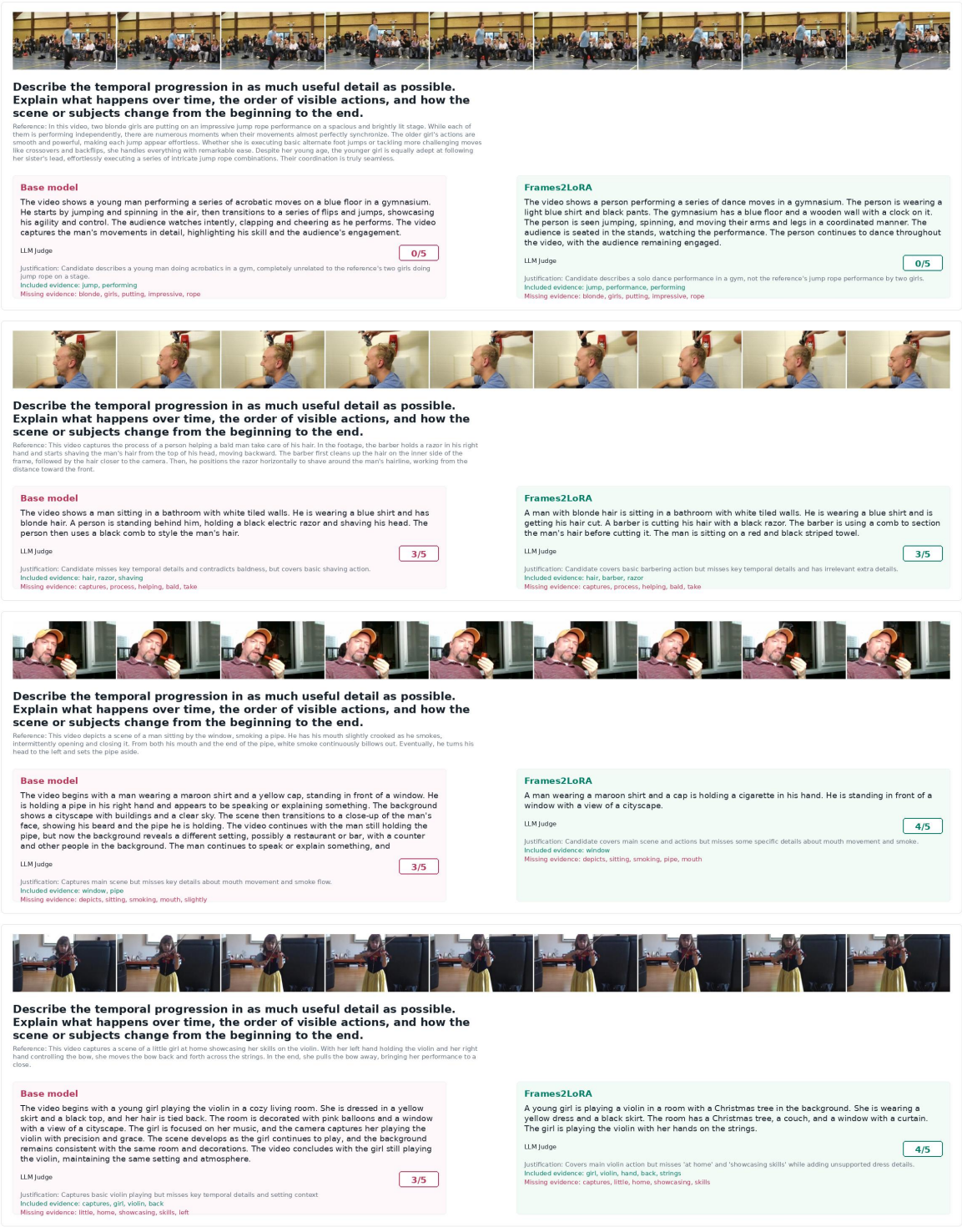
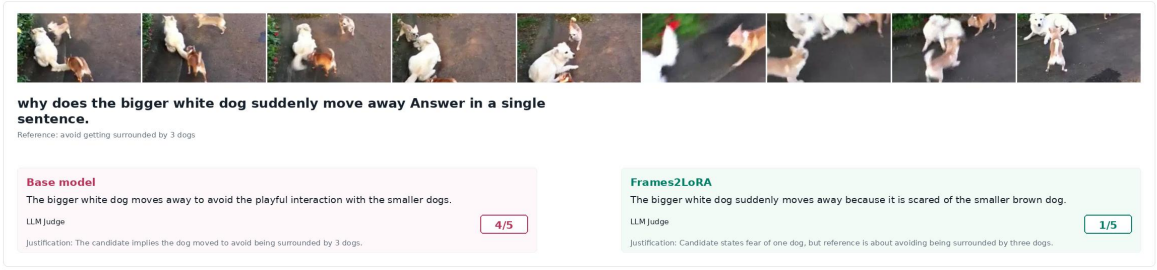


Figure 20: Qualitative examples from CaReBench: Temporal Caption.



why does the bigger white dog suddenly move away Answer in a single sentence.

Reference: avoid getting surrounded by 3 dogs

Base model

The bigger white dog moves away to avoid the playful interaction with the smaller dogs.

LLM Judge 4/5

Justification: The candidate implies the dog moved to avoid being surrounded by 3 dogs.

Frames2LoRA

The bigger white dog suddenly moves away because it is scared of the smaller brown dog.

LLM Judge 1/5

Justification: Candidate states fear of one dog, but reference is about avoiding being surrounded by three dogs.



why is woman displaying different vegetables to the children Answer in a single sentence.

Reference: teaching them about vegetable

Base model

To teach them about different vegetables and their uses in cooking.

LLM Judge 5/5

Justification: Candidate clearly implies teaching about vegetables, matching reference.

Frames2LoRA

The video shows a woman in an orange and white striped shirt and a white hat, wearing a chef's hat, standing in a kitchen with a red and white checkered floor. She is holding a small child dressed in a white shirt and a red and white striped hat. The woman is demonstrating how to use a small red and white toy chef's hat, which is placed on the child's head. The kitchen is equipped with a stove, a sink, and a countertop with various items, including a bowl of vegetables. The woman is likely teaching the child how to use the toy chef's hat in a playful and educational manner

LLM Judge 3/5

Justification: Candidate describes scene but misses 'different' vegetables from question.



why is there a piece of tissue in front of the baby Answer in a single sentence.

Reference: to wipe baby after feeding him

Base model

To keep the baby's head clean and dry.

LLM Judge 3/5

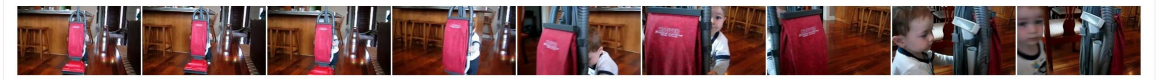
Justification: Candidate mentions cleaning but misses the specific purpose of wiping after feeding.

Frames2LoRA

The baby is being fed with a bottle of milk, and a piece of tissue is placed in front of the baby to prevent any spills or messes.

LLM Judge 4/5

Justification: The candidate implies the tissue's purpose is to prevent messes after feeding.



how did the boy move himself nearer to the camera Answer in a single sentence.

Reference: push the vacuum cleaner

Base model

The boy moved closer to the camera by walking towards it.

LLM Judge 3/5

Justification: Candidate describes walking but misses the specific action of pushing the vacuum cleaner.

Frames2LoRA

The video shows a young boy in a living room, pushing a red Hoover vacuum cleaner towards the camera.

LLM Judge 5/5

Justification: Candidate clearly implies the action of pushing the vacuum cleaner to move nearer.

Figure 21: Qualitative examples from NExT-QA.

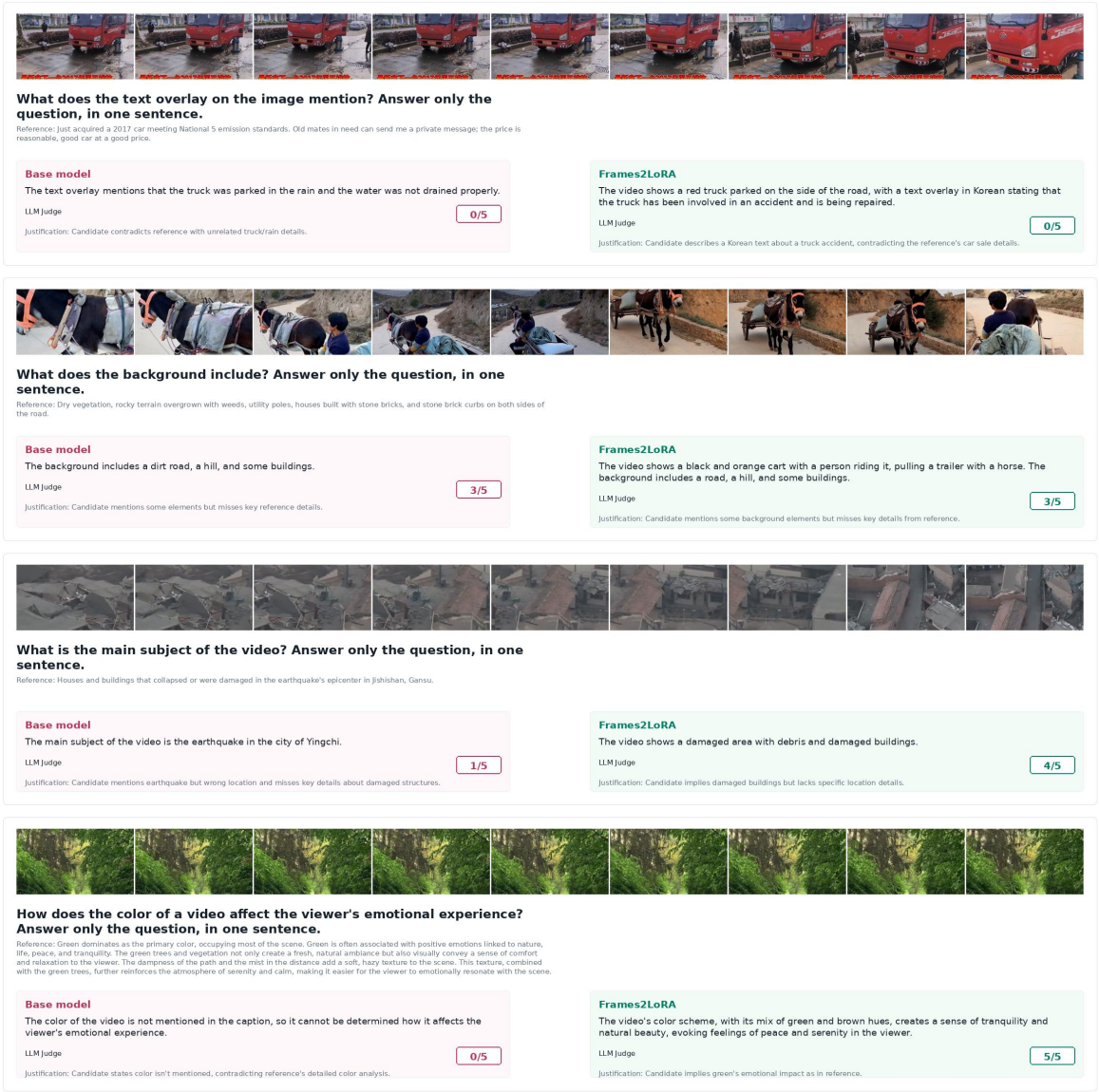


Figure 22: Qualitative examples from VidCapBench.

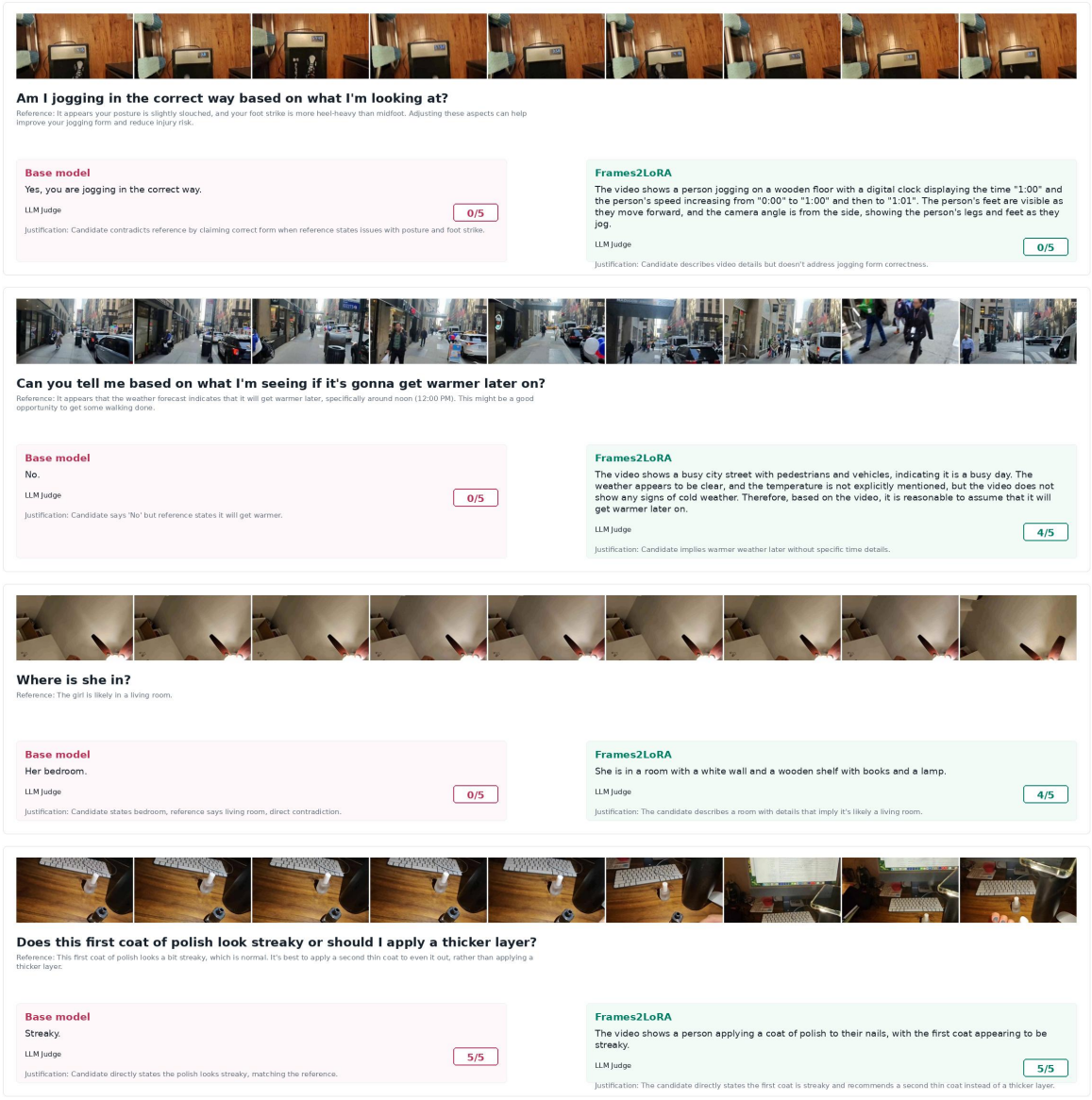


Figure 23: Qualitative examples from PLM SGQA.

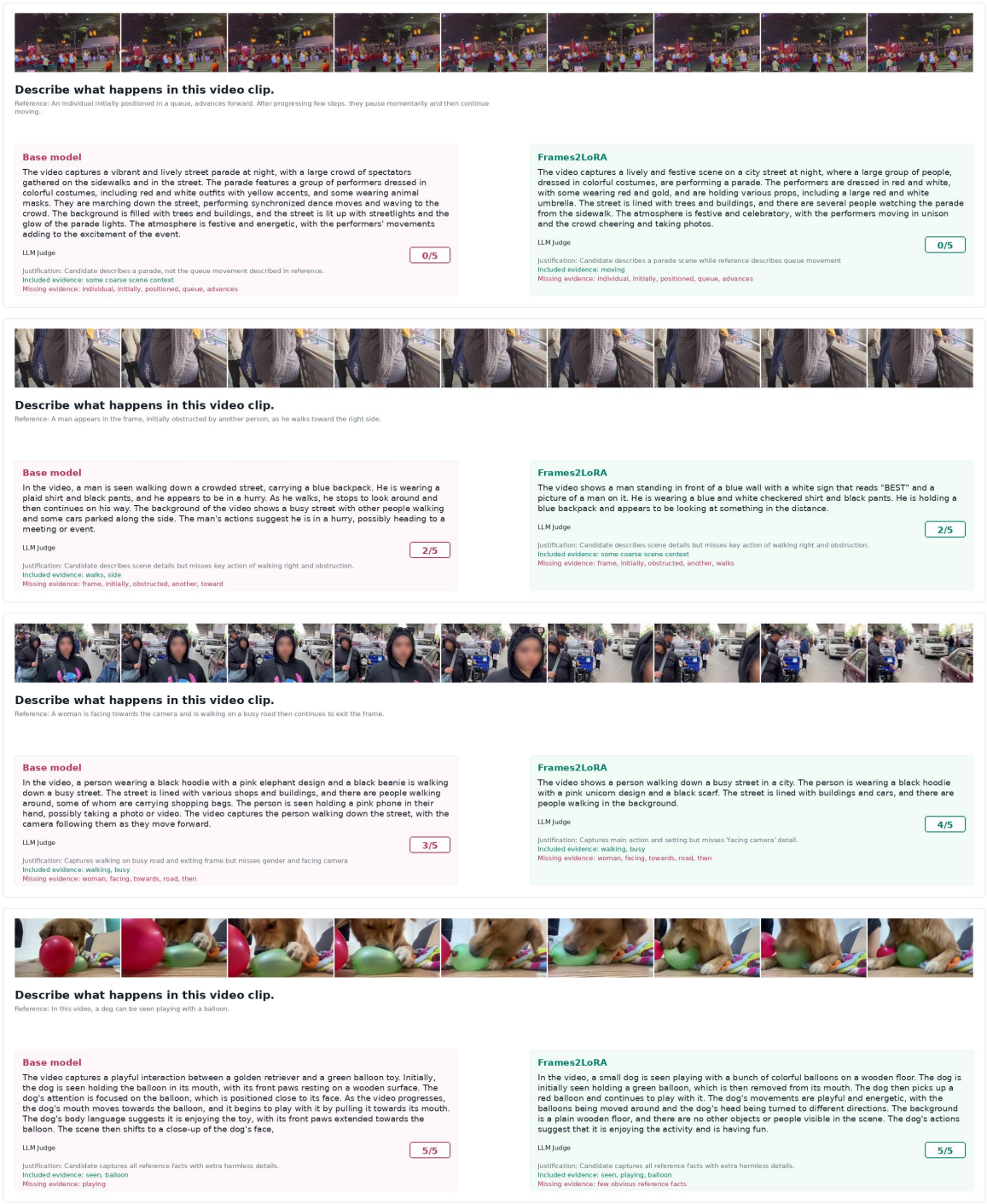


Figure 24: Qualitative examples from RCAP.

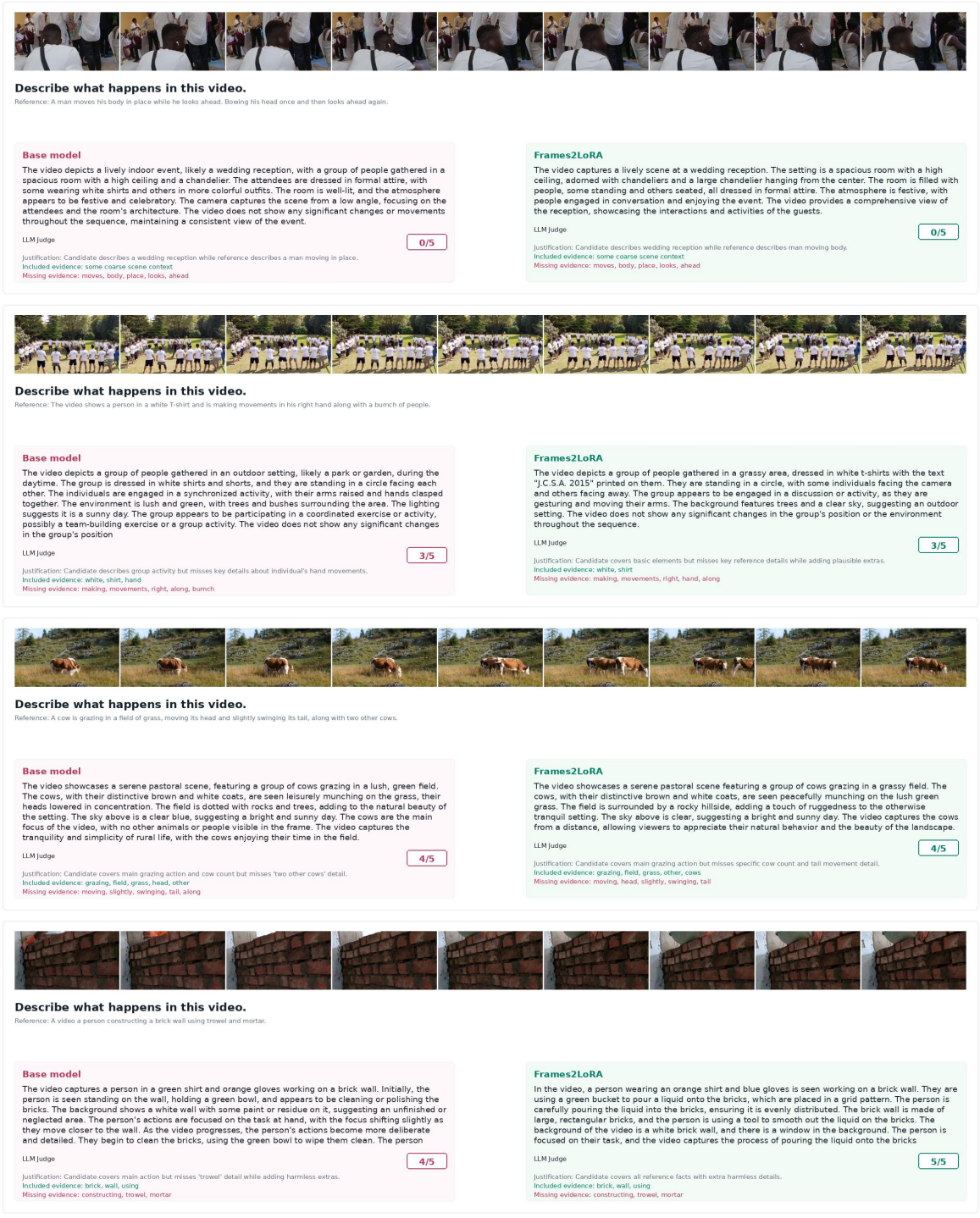


Figure 25: Qualitative examples from RDCAP.

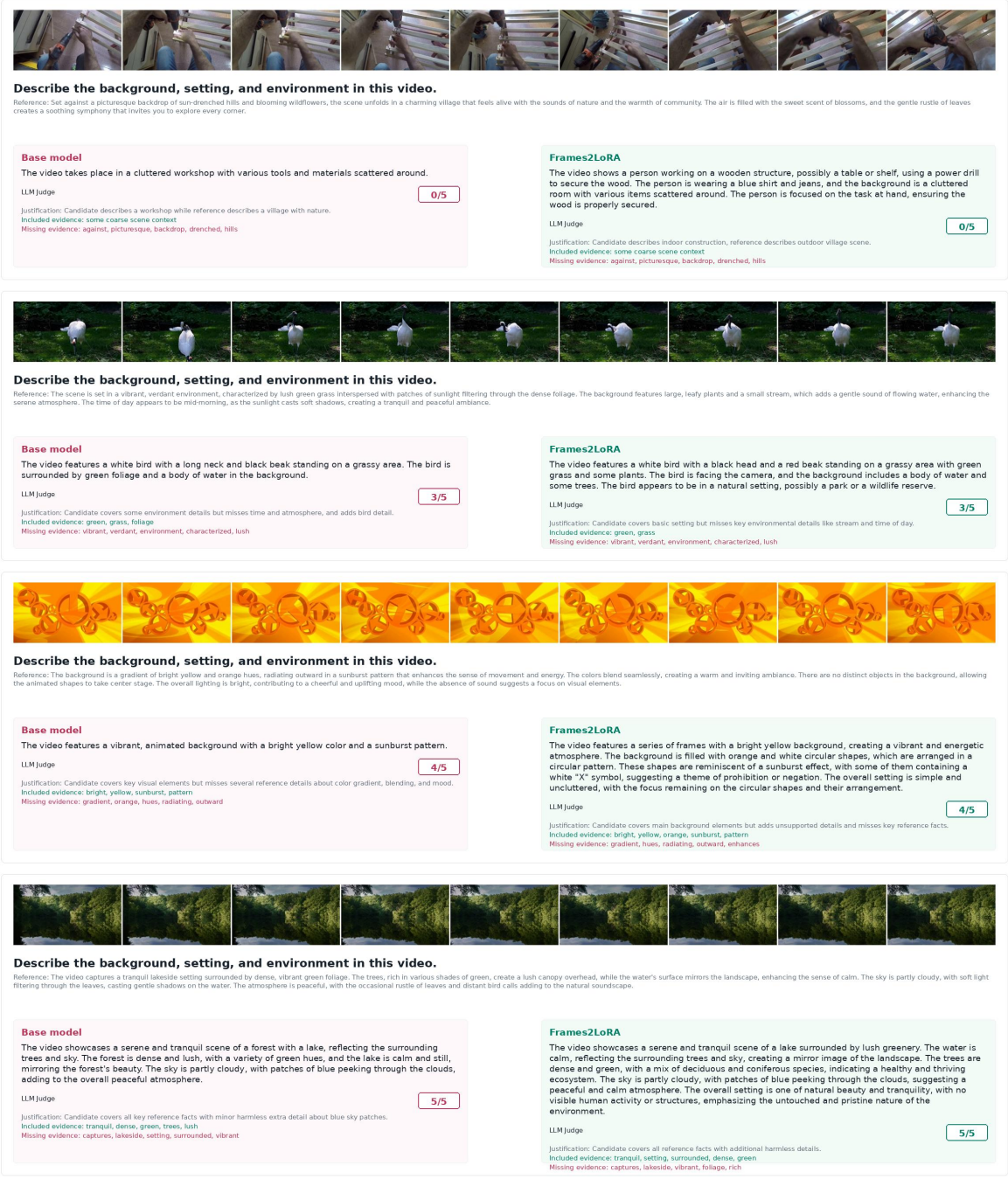


Figure 26: Qualitative examples from VDC Background.

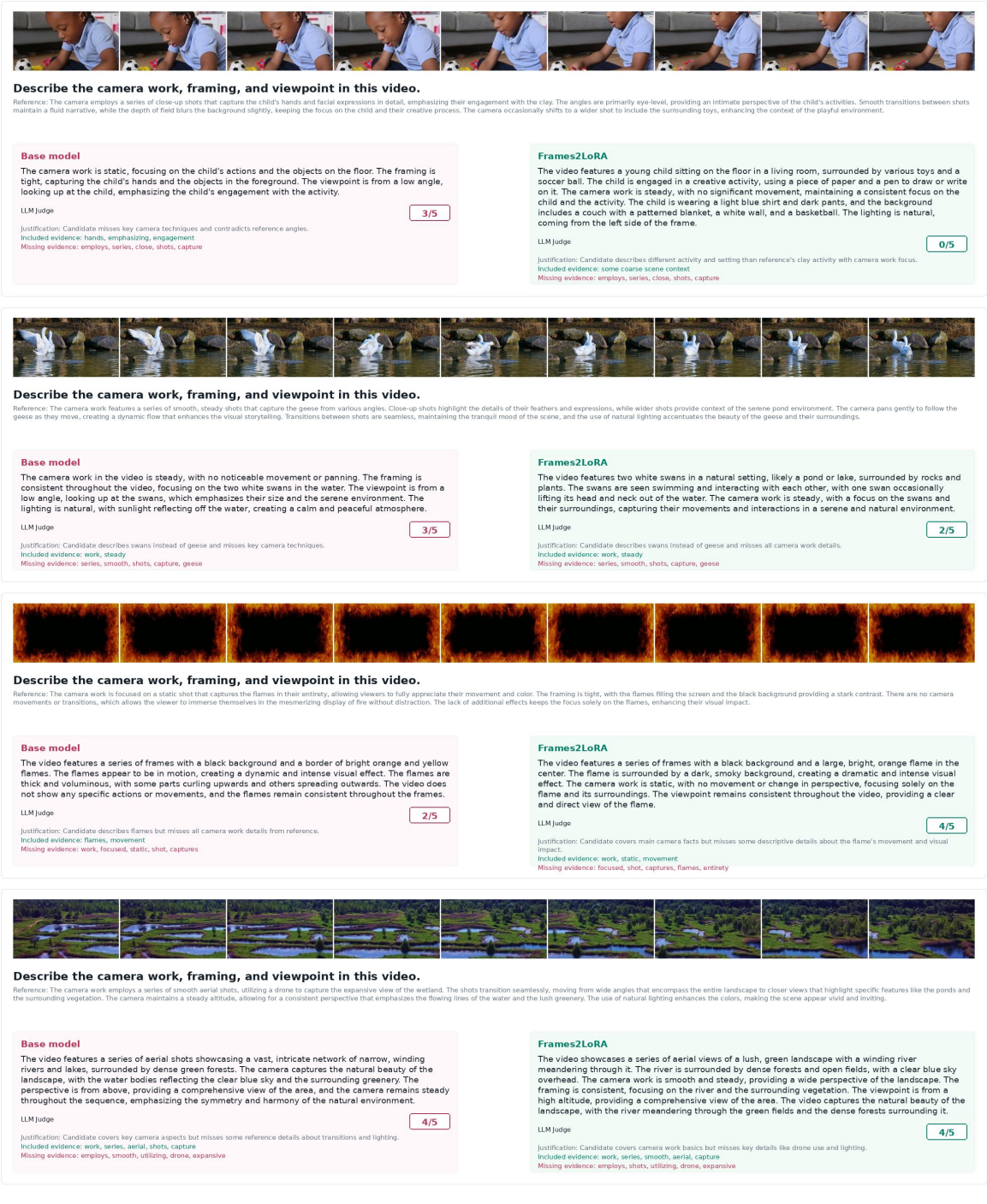


Figure 27: Qualitative examples from VDC Camera.

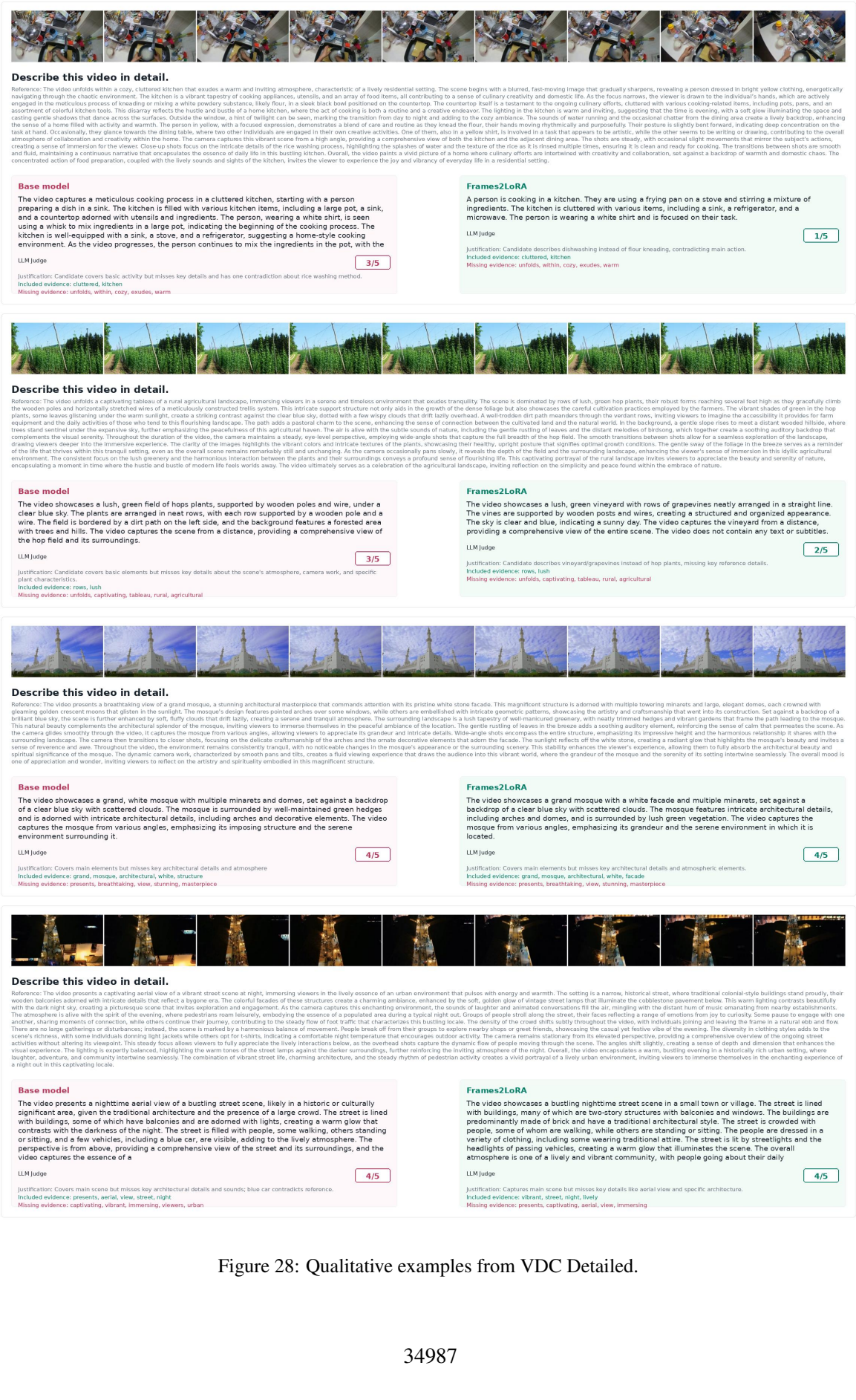


Figure 28: Qualitative examples from VDC Detailed.

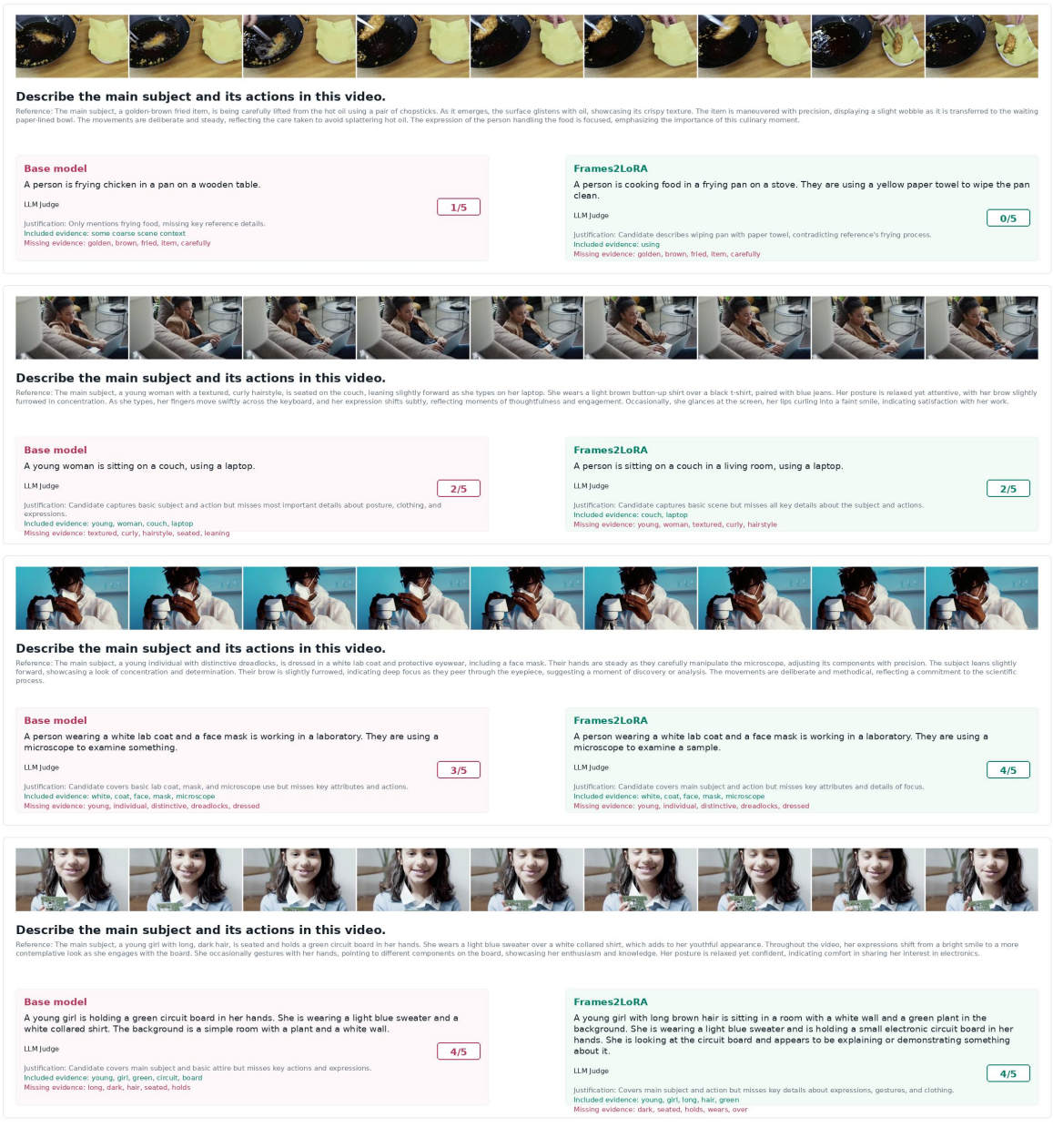


Figure 29: Qualitative examples from VDC Main Object.

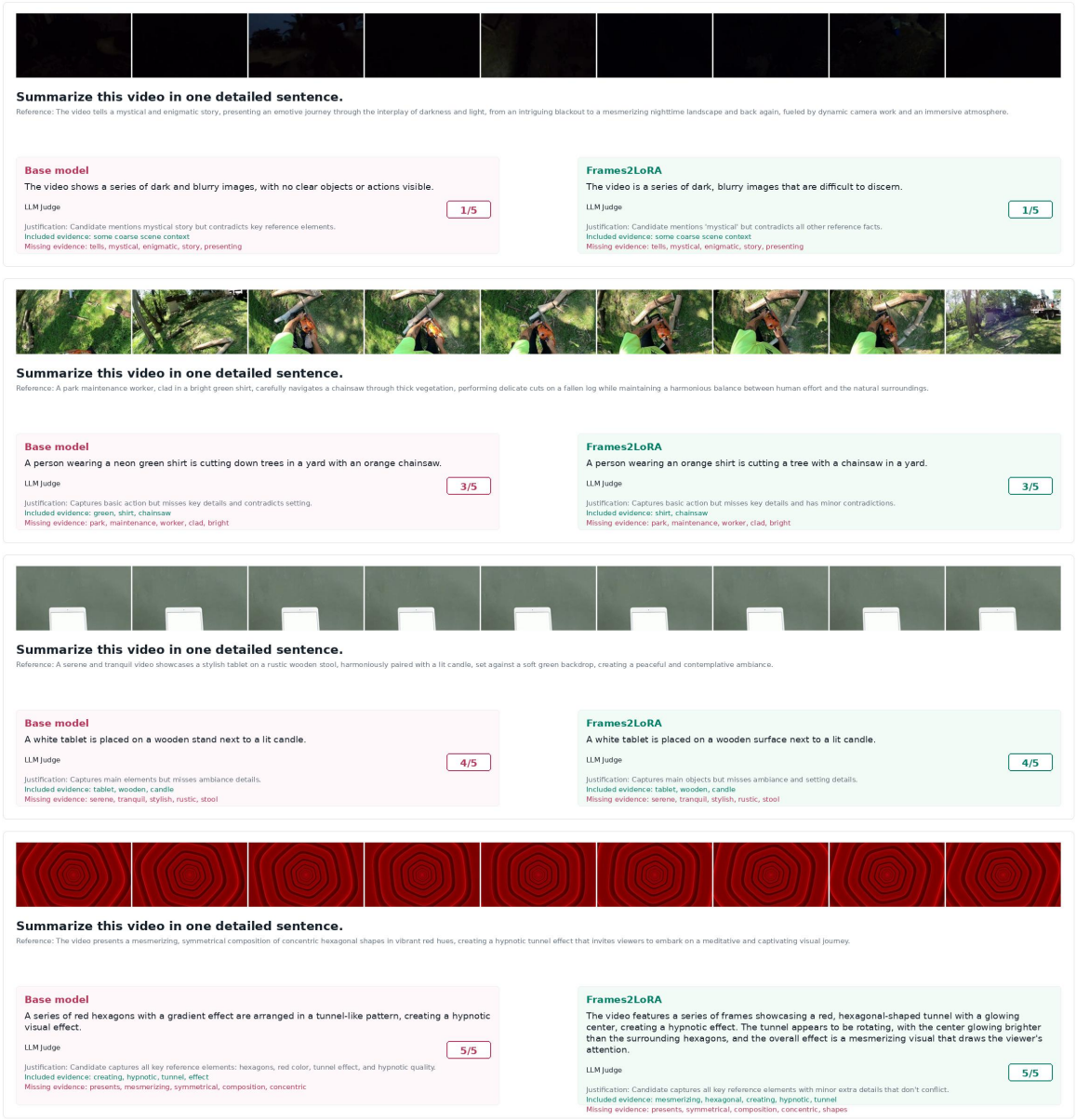


Figure 30: Qualitative examples from VDC Short.