Enhancing Video Vision-Language Models for Camera and Scene Understanding

Stephen Dong*
Cornell Tech
New York, NY, USA
sd2224@cornell.edu

Wuchen Li*
Cornell Tech
New York, NY, USA
wl758@cornell.edu

Audrey Tjokro*
Cornell Tech
New York, NY, USA
act245@cornell.edu

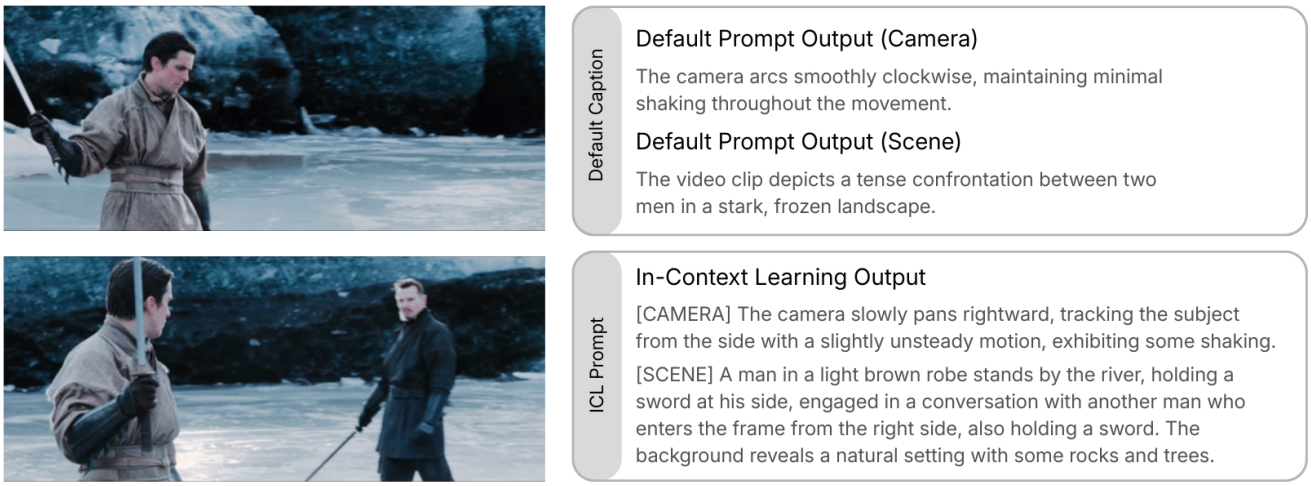


Figure 1: Overview of our setting: we probe how prompt design affects camera motion and scene understanding in video vision-language models.

Abstract

Video understanding tasks typically require models to capture both what is happening in a scene (semantic understanding) and how the camera moves to show it (cinematic understanding). Current video vision-language models (V-VLMs) tend to excel at one aspect while under-performing on the other. In this work, we investigate whether in-context learning (ICL) can help bridge this gap without additional fine-tuning.

We first identify and quantify a fundamental specialization trade-off: fine-tuning Qwen2.5-VL-7B on the CameraBench dataset improves camera motion understanding by 7.5% (BERTScore) but simultaneously degrades scene understanding by 4.85%. This reveals that camera and scene reasoning compete for model capacity.

We then systematically evaluate six ICL prompting strategies—including persona-based, chain-of-thought, curriculum, contrastive, and role-based approaches—designed to elicit both camera and scene descriptions simultaneously. Our experiments on 200 sampled CameraBench test videos show that ICL can locally rebalance the trade-off, but no prompting strategy achieves Pareto improvement over single-task baselines. This suggests the constraint is architectural rather than methodological.

We also uncover an important metric asymmetry: while BERTScore drops 1–5% under structured prompts, LLM-as-judge scores remain stable at 97–98%, indicating that BERTScore penalizes output format changes rather than semantic degradation. This finding has implications for evaluation methodology in prompt engineering research.

Our results suggest that achieving strong dual-task performance requires architectural innovations—not just prompting—to decouple geometric and semantic reasoning. We provide practical guidance for practitioners and validate our findings through a qualitative case study on medical ultrasound videos.

*Alphabetical order by last name. All authors contributed equally to this work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, Washington, DC, USA
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; *Computer vision*; *Machine learning*.

Keywords

video vision-language models; camera motion understanding; in-context learning; video captioning; multimodal learning

ACM Reference Format:
Stephen Dong, Wuchen Li, and Audrey Tjokro. 2025. Enhancing Video Vision-Language Models for Camera and Scene Understanding. In . ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

1.1 Limitations Camera-Motion Understanding in Video Vision-Language Models

Video Vision-Language Models (V-VLMs) exhibit strong semantic understanding of *what* appears in a scene (i.e. objects, actions, and narrative context), but struggle to capture cinematic understanding or *how* the scene is filmed. [3, 10]. More critically, real-world applications (medical video analysis, robotics, autonomous driving) require models to simultaneously understand both: the relationship between camera motion and scene geometry encodes 3D structure. When V-VLMs misinterpret camera geography *while maintaining* scene understanding, their spatial reasoning capabilities degrade. This is fundamentally a **dual-task problem**, not a single-task weakness.

Recent benchmarks aim to diagnose and solve this limitation. Among them, **CameraBench** [11] stands out as the first large-scale benchmark explicitly built to evaluate a model's ability to recognize and caption camera motion. Through the annotation of over 3,000 videos, the authors fine-tune open-source Qwen 2.5-VL-7B [1], on their dataset to show that this specialized model can outperform state-of-the-art proprietary V-VLMs such as GPT-4o [15] and Gemini-2.5-Pro [4] on camera motion captioning.

While fine-tuning is an effective solution, it poses key practical limitations . First, it requires access to model weights (which are unavailable for proprietary VLMs, which currently dominate performance). Second, it demands high-quality labeled video datasets, which are difficult and expensive to reproduce in specialized domains (e.g. health, robotics, or driving footage). Third, fine-tuned models are tied to a specific model snapshot and must be retrained in any case that a stronger base model become available. Finally, it is unclear whether these specialized fine-tuned models generalize broadly, as we later show that it introduces measurable tradeoffs in scene-level caption quality.

These limitations motivate us to propose an inference-side alternative. In this paper, we explore *in-context-learning* (ICL) as a strategy for improving camera-motion understanding in V-VLMs. Unlike fine-tuning, ICL operates purely through prompt design, bypassing the need for weight access or specialized datasets.

1.2 Proposed Apporach: Systematic Prompt Design for Dual Understanding

To investigate whether inference-time strategies can bridge the gap between camera and scene understanding without the drawbacks of fine-tuning, we conduct a systematic evaluation of In-Context Learning (ICL) strategies. Our approach departs from single-task optimization; instead, we design prompts that enforce a Dual-Task Constraint, requiring the model to disentangle and generate both cinematic (camera motion) and semantic (scene description) outputs within a single inference pass.

We construct a suite of six prompting strategies drawn from diverse ICL principles, including Persona-based prompting, Chain-of-Thought (CoT) reasoning, and Contrastive examples. Crucially, these prompts impose strict structural constraints (e.g., specific tags, line limits, and distinct "roles") to test not only the model's visual understanding but also its ability to maintain instruction adherence under the cognitive load of a dual objective.

To rigorously evaluate this approach, we map these ICL strategies against single-task baselines using a complementary metric framework:

(1) BERTScore: To measure semantic similarity against ground truth, highlighting potential length mismatches and formatting penalties.
(2) LLM-as-a-Judge: To evaluate the actual semantic quality and instruction compliance, allowing us to detect when models fail to generate required outputs (e.g., ignoring scene descriptions) despite high theoretical capability.

This setup enables a granular analysis of the "Specialization Trade-off," quantifying how prompt design influences the balance between geometric precision and semantic richness

1.3 Contributions

(1) **The Specialization-Compliance Trade-off:** We identify a critical side effect of specialized fine-tuning. While fine-tuning on CameraBench improves motion captioning by 7.5%, it induces a form of catastrophic forgetting in instruction following. We also find that the specialized model ignores scene description prompts entirely in test cases, revealing that standard fine-tuning compromises the model's ability to handle multi-objective constraints.
(2) **Structured ICL for Joint Camera–Scene Understanding:** We systematically evaluate six prompting strategies designed to elicit **dual-task** outputs (both camera and scene in one caption) rather than single-task specialization. We map these strategies across a 2D trade-off space to understand their efficacy and limitations.
(3) **Prompting's Limits and the Case for Architecture:** We demonstrate that prompting can reshape the trade-off curve locally but cannot achieve a Pareto improvement over single-task reference points. This reveals that architectural or training-time innovations— not just prompting—are necessary for balanced dual-task performance.
(4) **Methodology: Metric Asymmetry in Video Captioning:** We show that BERTScore and LLM-as-judge capture fundamentally different quality dimensions. For prompt engineering work, judges are more appropriate than reference-based metrics for assessing semantic quality despite format changes.

2 Related Work

2.1 Vision-Language Models and Video Understanding

Vision-language models have rapidly evolved from image-focused architectures (e.g., CLIP [16], LLaVA [12]) to video-capable systems capable of processing temporal sequences of frames or compressed video embeddings. Models such as Qwen-VL, GPT-4o, and Gemini-2.5 now handle extended video contexts, enabling rich semantic understanding through multi-modal pretraining on large-scale image-text and video-text corpora. These V-VLMs have demonstrated to excel at *semantic understanding* tasks, which includes recognizing objects, actions, spatial relationships, and narrative context. However, a growing body of literature suggests that V-VLMs struggle with *geometric and temporal precision.* These signals include cinematic elements such as shot size, framing, angle, and especially camera motion. Beyond capturing potential artistic intent, these signals are important from a practical standpoint because V-VLMs are increasingly deployed in sensitive domains (i.e. surgical videos, robotics, autonomous motion) [9, 21, 24]. Camera Motion also encodes 3D structure and Scene Geometry. If V-VLMS are unable to understand camera geography, it's entire spatial reasoning stack collapses. This asymmetry arises partly from training data imbalance: web-scale datasets contain abundant scene descriptions but sparse, precise cinematographic annotations [11]. Consequently, models learn strong language priors for semantic content but weaker geometric primitives for camera motion.

Recent work has attempted to bridge this gap through specialized datasets and fine-tuning. Architecturally, some approaches employ explicit geometric reasoning modules, dense frame sampling, or optical flow integration to improve motion estimation. Yet, the trade-off persists: gains in motion understanding often come at the cost of scene comprehension.

2.2 CameraBench and Limitations to Fine-Tuning Approaches towards Camera Motion Understanding

The gap in capturing the cinematic cues motivated the development of benchamrks aimed at measuring the cinematic and geometric understanding [3, 7, 10]. Among these, the **CameraBench** dataset, stands out as the first large-scale benchmark explicitly focused to on camera-motion understanding. Lin et al. constructed a dataset that comprises approximately 3,000 diverse internet videos annotated with a comprehensive taxonomy of camera motion primitives developed in collaboration with cinematographers. This taxonomy spans three reference frames (object-centric, ground-centric, camera-centric) and includes motion types, steadiness characterization, translation (dolly, pedestal, truck), rotation (pan, tilt, roll), intrinsic changes (zoom), and complex motions (arc, tracking).

Crucially, the CameraBench annotation framework employs a "label-then-caption" approach: annotators first classify motion primitives, then provide natural language descriptions to capture ambiguous or conflicting movements. This dual representation enables evaluation across both structured (classification) and open-ended (generation) tasks.

Evaluations on CameraBench reveal a critical finding: Structure-from-Motion (SfM) methods excel at geometric primitives (e.g., recovering zoom from perspective changes) but fail to capture semantic primitives that depend on scene content (e.g., recognizing a "tracking" shot requires detecting the moving subject). Conversely, V-VLMs struggle with geometric precision, but leverage semantic understanding. The authors also fine-tune an open-source model (Qwen-2.5Vl-7B) on their high quality dataset to demonstrate that it is capable of generating more accurate camera motion captions than the at the time state-of-the-art generative V-VLMs such as GPT-4o and Gemini-2.5-Pro.

However, fine-tuning as a general solution faces several critical limitations. Notably, it requires access to model weights, and while closed-sourced models (which currently stand as the strongest models), provide access to fine-tuning capabilities for text-based models, they do not largely allow fine-tuning to this date on V-VLMs.[4, 15] This essentially limits fine-tuning V-VLMs on the much weaker open-source models. Additionally, fine-tuning relies on high-quality supervised data, which is costly to produce and especially difficult to replicate for specialized domains (e.g. Medical or Robotics Videos). Fine-tuned models also suffer from version-lock as updates to the base model would require re-training. Finally, and perhaps more critically, it remains unclear whether models fine-tuned on cinematic internet videos generalize effectively to domain specific camera distributions. These limitations motivate inference-time, such as in-context learning, which do not require modifying the model weights.

2.3 In-Context Learning and Prompting for Multimodal Models

In-context learning (ICL) [5] refers to the ability of models to adapt their behavior based on the examples provided on the prompt, without explicit training or fine-tuning. For language models, ICL has been extensively studied: few-shot examples demonstrate improved performance on classification, generation, and reasoning tasks. The mechanisms underlying ICL remain an active research area; proposed explanations range from implicit task learning to activation of latent task-specific features.

Recent work has extended ICL to multimodal models. Studies demonstrate that multimodal foundation models (e.g., GPT-4o, Gemini 1.5 Pro) benefit from few-shot examples and can scale to many-shot scenarios (hundreds to thousands of examples) within long-context windows. Importantly, open-weight models like Llama 3.2-Vision and InternLM-XComposer2.5 show less pronounced ICL gains, suggesting that strong language priors and scale are important factors.

For video-language tasks, ICL strategies remain relatively understudied, especially in specialized domains like camera motion understanding. Prior work on video captioning and visual question-answering shows that few-shot examples improve accuracy, yet systematic studies of ICL design principles (e.g., example selection, prompt framing, task decomposition) in the video domain are limited. Unlike fine-tuning, ICL modifies only the input prompt, raising important quesitons about how to best evaluate its effects. Because prompt structure can alter output format and lexical choices, evaluation requires both reference-based and reference-free metrics.

2.4 Evaluation Metrics for Video Captioning

Evaluating video captioning requires balancing reference-based metrics (that compare generated captions to ground truth) with reference-free metrics and human judgments. Common reference-based metrics include BLEU, ROUGE, and METEOR, which rely on n-gram overlap; these are efficient but often correlate poorly with human judgments for open-ended tasks where multiple valid descriptions exist.

BERTScore [23] addresses this limitation by computing token-level semantic similarity using contextual embeddings from pre-trained language models (e.g., BERT, RoBERTa). Rather than exact word matches, BERTScore computes cosine similarity between token embeddings, making it more robust to paraphrasing. For camera motion descriptions, where valid captions can express the same motion in varied linguistic ways, BERTScore provides a more appropriate metric than n-gram overlap.

LLM-as-Judge [6] represents an alternative evaluation paradigm: using a large language model (or multimodal model) to assess caption quality according to specified criteria. An LLM judge can evaluate consistency, relevance, factual accuracy, and completeness, and can assign scores, comparative rankings, or detailed feedback. While LLM judges introduce potential biases and hallucinations, they offer flexibility and often correlate closely with human preferences on subjective tasks.

Recent work has explored both strengths and limitations of LLM judges in vision-language tasks, finding that they perform well on pair comparisons but exhibit greater variance on absolute scoring. We employ both metrics, BERTScore for reference-based evaluation and LLM-as-judge for holistic quality assessment, to provide complementary perspectives.

2.5 Applications in Medical and Surgical Video Understanding

A growing body of work studies computer vision and language models for surgical and medical video analysis, focusing on tasks such as surgical phase recognition, tool detection, workflow assessment, and skill evaluation. Traditional approaches rely on domain-specific supervised models trained on labeled datasets, which are expensive and time-consuming to curate.

Recent work has explored zero-shot and few-shot video-language models (e.g., GPT-4o, Qwen2.5-VL) for surgical video analysis, finding that general-purpose models can perform rudimentary surgical report generation and object detection but struggle with fine-grained procedural understanding or grading pathology. Importantly, few-shot in-context learning has been shown to improve surgical video understanding, helping models avoid confusing instrument motion with camera motion and generating more medically coherent scene descriptions.

Our work is motivated in part by these clinical applications: in surgical video, stable camera understanding coupled with precise scene description is critical for educational purposes, surgical skill assessment, and intraoperative decision-support. Privacy constraints and data scarcity make custom model development challenging, making accessible prompting strategies particularly valuable.

3 Dataset and Preprocessing

We utilize the CameraBench benchmark as our primary source for video data and camera motion labels. The testing dataset comprises 1071 diverse videos sourced from the internet, annotated by experts with a comprehensive taxonomy of camera motion primitives (e.g., pan, tilt, zoom, dolly). These annotations serve as our ground truth for evaluating the "Geometric" dimension of the dual-task trade-off.

3.1 Video Processing Pipeline

As CameraBench videos are formatted initially as GIFs, we convert each GIF to MP4 using the *imageio-ffmpeg* backend, which invokes *ffmpeg* with H.264 encoding and a fixed frame rate to ensure consistent frame sampling across models. We normalize frames to to RGB and uniformly sample 32 frames at 8 fps from each clip. We apply this preprocessing across all models and prompting conditions, which ensures that performance differenses arise from model or prompt rather than the input variation.

3.2 Scene Annotation Pipeline

Because CameraBench does not provide ground-truth scene descriptions suitable for evaluation, we construct a reference set of scene captions. For each clip, we query Gemini 2.5 Pro with a single deterministic prompt, "Describe the scene in detail." We run Gemini at Temperature 0 to produce stable, reproducible captions. These captions serve exclusively as reference descriptions for evaluation and are not used for model training/finetuning.

4 Methods

4.1 Models

We evaluate two versions of Qwen2.5-VL-7B and use Gemini 2.5 Pro as reference.

4.1.1 Qwen2.5-VL-7B (pre-trained). We use the pre-trained Qwen2.5-VL-7B instruct model as our baseline for general-purpose multimodal understanding. This model has not been exposed to the specific CameraBench training set, serving as the proxy for "balanced" but non-specialized performance.

4.1.2 Qwen2.5-VL-7B (CameraBench-Fine-Tuned.) This variant was fine-tuned specifically on the CameraBench training split using the geometric camera motion annotations. It represents the "Specialized" condition in our trade-off analysis. While it achieves state-of-the-art performance on camera motion captioning, our experiments investigate whether this specialization comes at the cost of semantic instruction following. Inference settings are kept identical to the pre-trained model to ensure a fair comparison.

4.1.3 Gemini 2.5 Pro. This is a large foundation VLM accessed via the Gemini API. We use Gemini for two purposes:
(1) A modern reference point for reproduction analysis, and
(2) an annotater for constructing scene-level ground-truth captions.

4.1.4 Inference Configuration. We standardize the video input and text-generation settings across all Qwen experiments.

- **Decoding:** Greedy decoding (no sampling or beam search)
- **Maximum Output Length:** 256 new tokens for all prompts.
- **Precision:** FP16 inference on a single GPU, no quanitzation.

For scene level annotations, Gemini 2.5 Pro is queried at temperature 0.

4.2 Caption Generation

For each clip, we generate captions by issuing a single forward pass of the model under each prompting condition (default and ICL variants), using identical inference parameters to isolate the effect of the prompt itself. Qwen2.5-VL produces both camera-motion and scene captions directly, while Gemini 2.5 Pro is used only to produce deterministic reference scene captions at temperature 0. All model outputs are collected verbatim without any post-processing or filtering, and are used exactly as generated for downstream evaluation.

4.3 Dual-Task Setting and Reference Points

Our experimental setting departs from single-task specialization: rather than optimizing for camera motion alone, we evaluate models on their ability to handle both camera motion and scene understanding. To establish reference points for this dual-task space, we define two single-task baselines:

- **Camera-only baseline:** Pre-trained or fine-tuned Qwen with prompt "Describe the camera motion in this video." This represents the upper bound for pure motion understanding.
- **Scene-only baseline:** Pre-trained Qwen with prompt "Describe the scene in detail." This represents the upper bound for semantic scene understanding.

These single-task baselines are not our proposed solution; they define the frontier. Our ICL prompts, by contrast, are constrained to produce both outputs simultaneously, making them inherently more challenging and more practical for real-world applications.

4.3.1 Default Prompting (Baseline Evaluations). To generate baseline evaluations for how Qwen2.5-VL-7B and Qwen2.5-VL-7B (SFT) perform across the camera motion and scene recognition tasks, we generate captions using simple prompts. For camera motion, we use the same prompt as specified wihtin the original paper [11]: "Describe the camera motion in this video". For scene context, we use a prompt in similar style "Describe the scene in detail." By staying consistent with the original paper, this baseline also serves as a **replication** to the original CameraBench's experiment.

4.3.2 ICL Prompting (Our Extension). To study whether in-context learning improves video understanding, we nine in-context learning (ICL) prompting strategies to test whether structured prompt design improves video caption performance. These prompts are intended to capture both Camera Motion and Scene Recognition in order to try to achieve a balance via ICL. Each strategy introduces different cognitive prior or constraint for the model. We summarize the nine strategies (we include the full templates in the Appendix)

(1) **Persona ICL (Two-Line Role Prompt)** This strategy frames the model as an *expert video analyst* and enforces a rigid two-line output format separating camera motion from scene description. The persona framing is motivated by work showing that assigning explicit roles or personas can steer large language models toward more consistent, task-aligned behavior, especially for fine-grained generation tasks.[22] The strict line-level structure is designed to reduce camera–scene entanglement errors while keeping outputs short and evaluation-friendly.
(2) **Persona ICL (Paragraph/Tagged Prompt)** This strategy keeps the same high-level persona (*professional video captioner*) but relaxes the format into a single short paragraph with two tagged clauses: `[CAMERA]` and `[SCENE]`. The goal is to test whether a slightly more natural prose structure preserves the benefits of role conditioning while reducing over-constrained phrasing. Prior work on role-based and decomposed prompting suggests that separating sub-tasks into labeled segments can improve factuality and adherence to constraints, even when the output is more free-form. [22]
(3) **Chain of thought ICL** The third strategy explicitly encourages internal reasoning by asking the model to first "silently think" about camera motion and scene content, then compress that reasoning into two short sentences: one for `CAMERA_MOTION` and one for `SCENE_DESCRIPTION`. This follows chain-of-thought prompting work showing that encouraging intermediate reasoning steps can improve alignment on complex, compositional judgments. [20] By forcing a final concise summary after hidden reasoning, this prompt probes whether CoT-style setups help the model better disentangle motion and scene semantics without sacrificing brevity.
(4) **Curriculum ICL** This strategy orders exemplars from easy to medium to hard. [26] The motivation is to test whether gradually increasing motion complexity helps model better separate and articulate camera-motion behavior while maintaining accurate scene descriptions.
(5) **Contrastive ICL** This strategy contains both 'GOOD' and 'BAD' examples of camera-motion and scene descriptions. 'GOOD' examples cleanly separate motion from the scene context, while 'BAD' examples intentionally mix the two. the model is instructed to imitate the 'GOOD' examples only. This follows contrastive ICL findings that suggest that explicit negative examples improve adherence to constraints. [8]
(6) **Role-Based ICL** Inspired by persona and role-based prompting work, this strategy decomposes the task into two cooperative roles : a *camera operator* that describes only camera motion and a *scene observer that only describes scene content.* This framing encourages the model to disentangle camera-motion reaosning from semantic scene understanding before producing its output. [25]

4.4 Evaluation

We evaluate model outputs using two complementary metrices (1) automated text-text similarity measures (BERTScore), and (2) model-based judging metrics (LLM-as-a-judge). These evaluations are computed separately for the camera-motion task (which includes the expert-provided gold-labels) and the scene-description task (for which we construct the reference set using Gemini 2.5 Pro as decsribed above).

4.4.1 BERTScore. We compute BERTScore (F1-variant) using RoBERTa-large embeddings. BERTScore is computed for each caption-reference pair using a batch size of 32, with identical settinggs across all prompting conditions. BERTScore evaluates captions by comparing

contextual embeddings as opposed to raw word overlap (that n-grams based metrics capture). This makes it robust to paraphrasing, which is an important property in camera motion descriptions.

4.4.2 LLM-as-Judge. To complement the embedding-based similarity scores, we also employ an LLM-as-a-judge procedure. We leverage Gemini 2.5 Pro, in which for each candidate-reference pair, we construct a fixed prompt that presents both captions and asks:

> "Does the candidate caption convey the same meaning as the reference caption? Answer 'Yes' or 'No' and provide your confidence."

Gemini returns a short natural language response such as "Yes (85%)", which we parse into (1) a binary decision, and (2) a numeric confidence score [0, 1] using a regex-based extraction method. The final evaluation method in this case is a confidence for the Yes decision. Unlike BERTScore, this judge-based score is sensitive to fine-grainted semantic errors. For example. LLM-as-a-judge might penalize more on getting the camera motion direciton wrong (e.g. pan left vs pan right) while these errors can look lexically similar but represent fundamental different actions.

5 Experiments

5.1 Experimental Setup

- **Dataset:** CameraBench test split (N = 1,071 video clips). [11]
- **Models Evaluated:** Qwen2.5-VL-7B (pre-trained) and Qwen2.5-VL-7B (fine-tuned on CameraBench). [1]
- **Prompting Conditions:** Baseline (default prompt) vs. ICL-enhanced prompts (multiple strategies).
- **Evaluation Metrics:** BERTScore [23] (F1, Precision, Recall), LLM-as-judge [6]
- **Sample Size:** Preliminary experiments on 192–200 videos; final results averaged over the full test split.

For each configuration we report two BERTScores: one for camera motion and one for scene description. For the single-task reference points, these scores come from separate runs with camera-only and scene-only prompts. For the dual-task ICL setups, both scores are computed from the same caption output, which contains both camera and scene descriptions.

5.2 Dual-Task Setting and the Trade-off Frontier

All experiments operate within a dual-task framework: models are evaluated on their ability to produce **both** accurate camera-motion descriptions **and** accurate scene descriptions for the same video clip. This is fundamentally different from optimizing for camera motion alone.

To characterize the trade-off landscape, we establish two single-task reference points:

(1) **Fine-tuned Default (Camera-Specialized):** Qwen2.5-VL-7B fine-tuned on CameraBench with camera-only prompt. Achieves 0.9284 camera BERTScore.
(2) **Pre-trained Default (Scene-Strong Reference):** Pre-trained Qwen2.5-VL-7B evaluated with a camera-motion prompt (0.8636 camera BERTScore) and a separate scene-description prompt (0.8728 scene BERTScore). Together these two single-task scores act as a reference for balanced scene understanding.

Model Variant	Prompt	Camera BERTScore	Scene BERTScore	Camera Δ (%)	Scene Δ (%)
Pre-trained	Default	0.8636	0.8728	0.00	0.00
	Persona 2-Line	0.8551	0.8473	-0.98	-2.92
	Persona Para/Tag	0.8570	0.8599	-0.76	-1.48
	Chain of thought	0.8546	0.8458	-1.04	-3.09
	Curriculum	0.8529	0.8436	-1.24	-3.35
	Contrastive	0.8528	0.8426	-1.25	-3.46
	Role-Based	0.8518	0.8450	-1.37	-3.19
Fine-tuned	Default	0.9284	0.8305	+7.50	-4.85
	Persona 2-Line	0.8850	0.8316	+2.48	-4.72
	Persona Para/Tag	0.8917	0.8357	+3.25	-4.25
	Chain of thought	0.8907	0.8252	+3.14	-5.45
	Curriculum	0.8808	0.8295	+1.99	-4.96
	Contrastive	0.8828	0.8276	+2.22	-5.18
	Role-Based	0.8819	0.8292	+2.12	-5.00

Table 1: Dual-task performance across single-task reference points and ICL strategies. The two reference points (Plus signs: Camera-specialized Fine-tuned default at 0.9284 camera; Balanced Pre-trained default at 0.8636 camera, 0.8728 scene) define single-task upper bounds. Our ICL prompts (circles) are constrained to output both camera and scene descriptions, requiring navigation of the dual-task trade-off space.

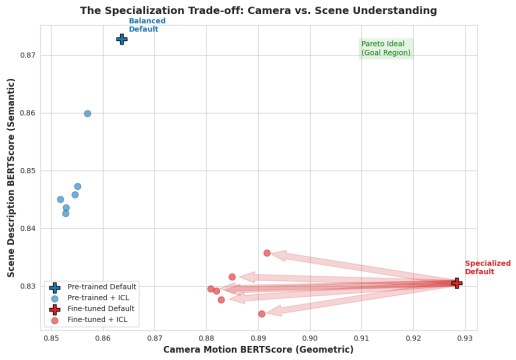


Figure 2: Visualizing the Specialization Trade-off. The two Anchors (Plus signs) define the frontier. ICL prompts (dots) move the Fine-tuned model (red) leftward, sacrificing camera precision for marginal scene gains, but fail to reach the "Pareto Ideal" region (top-right).

All ICL prompts are dual-task outputs: they must produce both camera and scene descriptions, placing them inherently between the single-task extremes.

5.3 The Specialization Trade-off and ICL Evaluation

We analyze the trade-off between camera motion and scene understanding using a unified 2D framework.

5.3.1 Unified Trade-off Results: Camera vs. Scene Performance.

Model Variant	Prompt	Camera LLM-Judge	Scene LLM-Judge
Pre-trained	Default	0.9813	0.9466
	Persona 2-Line	0.9779	0.9552
	Persona Para/Tag	0.9786	0.9508
	Chain of thought	0.9766	0.9539
	Curriculum	0.9798	0.9560
	Contrastive	0.9828	0.9585
	Role-Based	0.9788	0.9540
Fine-tuned	Default	0.9731	0.9749
	Persona 2-Line	0.9728	0.9645
	Persona Para/Tag	0.9756	0.9601
	Chain of thought	0.9772	0.9678
	Curriculum	0.9755	0.9663
	Contrastive	0.9778	0.9673
	Role-Based	0.9778	0.9655

Table 2: LLM-as-judge confidence scores show metric asymmetry: despite BERTScore declining 1%–5%, judges maintain 97%-98% confidence. This suggests judges capture holistic quality while BERTScore is overly sensitive to output format changes.

Interpreting the 2D Trade-off Space. Table 1 reveals three distinct regions in the 2D landscape:

(1) **Pre-trained + ICL (down-left region):** All prompts move down and left, degrading both camera (−0.98% to −1.37%) and scene (−1.48% to −3.46%). This suggests the pre-trained model's default behavior is already well-aligned with typical caption format; reformatting introduces friction.
(2) **Fine-tuned + ICL (middle region):** Prompts pull the fine-tuned model back toward the pre-trained baseline, losing 3.2%–5.0% of its original camera advantage (7.5%) while providing marginal scene improvements (+0.0% to +0.6%). These moves are *locally favorable* but do not achieve global Pareto improvement.
(3) **No Pareto-optimal solution:** Critically, no ICL prompt positions us at a point that simultaneously beats the fine-tuned camera score (0.9284) *and* the pre-trained scene score (0.8728). This reveals a fundamental constraint: the specialization trade-off cannot be overcome by prompting alone.

5.3.2 LLM-as-Judge Scores Reveal Metric Asymmetry. While BERTScore is sensitive to output formatting, LLM-as-judge provides a complementary holistic view. Despite BERTScore degrading by 1%–5% with ICL prompts, judges maintain 97%-98% confidence across nearly all conditions (Table 2).

Key Insight on Metric Complementarity. The disconnect between BERTScore and LLM-as-judge is striking and methodologically significant:

- **BERTScore penalizes format changes:** When output is restructured (e.g., adding labels or tags), token-level embeddings shift, causing BERTScore to drop even though semantic content is identical.
- **Judges assess holistic quality:** When evaluating whether candidate and reference convey the same meaning, judges correctly recognize that reformatted outputs preserve semantic equivalence.
- **Implication for research:** For prompt engineering studies, relying solely on reference-based metrics like BERTScore can mislead researchers about the true utility of structured prompts. Judge-based evaluation is more appropriate for this domain.

5.4 Summary: The Trade-off Frontier

Summarizing the quantitative results from Phase 2, we observe a distinct "Specialization Trade-off" that prompting alone fails to overcome.

(1) **Dual-task reasoning requires architectural trade-offs.** Fine-tuning specializes in camera motion (+7.5%) at the cost of scene understanding (-4.9%). This reveals that achieving **joint** camera–scene understanding is fundamentally constrained by model capacity, not a prompting failure.
(2) **ICL can rebalance locally but cannot transcend dual-task constraints.** Our prompts move the fine-tuned model toward better scene understanding (marginal +0.0%-+0.6%), but this comes at the cost of sacrificing camera precision (3.2%-5.0% degradation). Critically, prompting still cannot simultaneously achieve the camera-specialized upper bound (0.9284) *and* the scene-balanced upper bound (0.8728)—a dual-task impossibility, not a prompting weakness.
(3) **Prompting is not a substitute for architectural innovation.** To achieve a solution that beats both anchors—high camera precision *and* strong scene understanding—requires approaches beyond prompting: multi-task training objectives, architectural modifications that decouple geometric and semantic reasoning, or ensemble methods.

6 Qualitative Analysis: Ultrasound Video Case Study

To test whether our prompting strategies transfer beyond CameraBench, we conduct a qualitative case study on abdominal ultrasound instruction videos. Ultrasound is a clinically important setting where video understanding depends heavily on probe (camera) motion: sweeping, sliding, and fanning the probe determines what anatomy comes into view and whether trainees achieve adequate coverage [14, 18, 19].

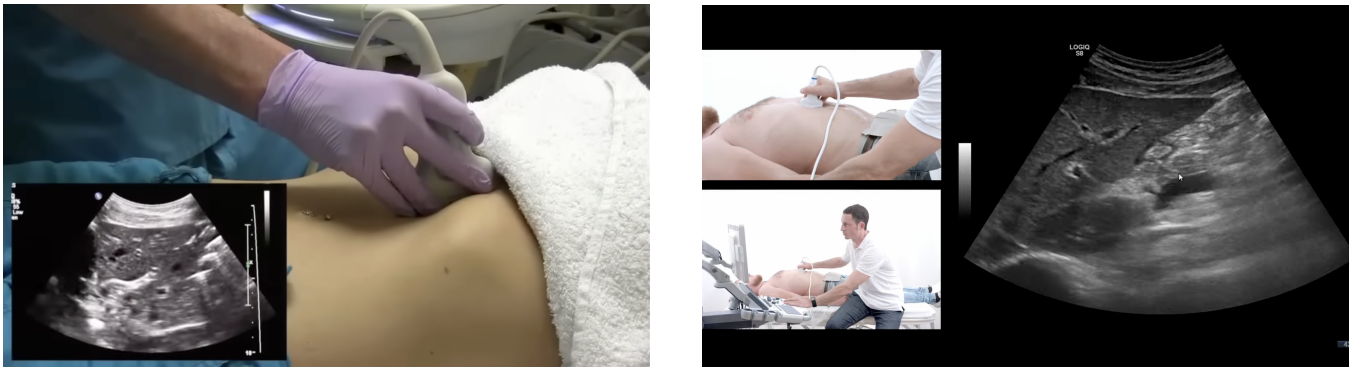
We use two publicly available teaching videos as representative examples: an *abdominal ultrasound* tutorial [17] and a *pancreatic ultrasound* tutorial [13] (both narrated physician-led demonstrations). These videos contain prolonged periods where the clinician steadily moves the probe while the ultrasound B-mode view changes in real time, making them ideal for evaluating camera motion versus scene understanding.

6.1 Visual Overview

Figure 3 shows representative frames from the two videos.

6.2 Case Study Design

For each video, we manually select short clips (5–10 seconds) where the clinician performs a controlled sweep or fan of the probe while



(a) Abdominal Ultrasound **(b) Pancreatic Ultrasound**

Figure 3: Representative frames from abdominal (left) and pancreatic (right) ultrasound teaching videos used in our qualitative case study. In both cases, probe (camera) motion and scene content must be disentangled to produce clinically meaningful captions.

narrating the target anatomy. We then generate captions with the model's **default prompt** (camera and scene entangled), and our best-performing **Structured Two-Line ICL prompt** which explicitly separates `CAMERA_MOTION` and `SCENE_DESCRIPTION`.

Because no human ultrasound caption dataset is available, we perform a qualitative, expert-motivated comparison rather than a full quantitative evaluation. We focus on three dimensions that matter for training and guidance systems [2, 18]:

(1) **Camera Motion Clarity:** Does the caption correctly describe how the probe (camera) is moving (for example sliding, fanning, angling) rather than organ motion?
(2) **Scene Specificity:** Does the scene description mention relevant organs and landmarks (for example liver, portal vein, pancreas) instead of vague phrases?
(3) **Role Separation:** Are camera motion and anatomical content kept conceptually separate, enabling downstream systems to reason over each dimension independently?

6.3 Qualitative Results

Table 3 shows representative outputs for one abdominal clip and one pancreatic clip. The default prompt tends to mix probe handling and anatomy into a single sentence, occasionally attributing motion to the organ instead of the camera. In contrast, the Structured Two-Line ICL prompt yields cleaner decomposition: camera motion focuses on how the probe is swept, while scene descriptions emphasize organ configuration.

Across several clips from both videos, we observe consistent patterns:

- **Cleaner motion semantics.** Default prompts sometimes describe organs "moving" or the image "shifting" without clarifying that the motion originates from the probe. ICL prompts more reliably label the camera motion as sliding, fanning, or tilting.
- **More anatomical detail.** When forced to reserve one line for scene content, the model tends to mention specific structures (for example liver, pancreas, diaphragm, vessels) instead of generic "organs" or "abdominal area".

Video	Prompt	Representative Output
Abdomen	Default	*"The ultrasound view moves slowly across the upper abdomen, showing the liver and nearby organs as the image slides back and forth."*
Abdomen	Structured Two-Line ICL	*CAMERA_MOTION: The probe slides smoothly from right to left under the costal margin in a steady sweep. SCENE_DESCRIPTION: The liver parenchyma and diaphragm come into view, with adjacent vessels and soft tissue visible in the upper abdomen.*
Pancreas	Default	*"The pancreas area shifts around on the screen as the ultrasound wand moves over the stomach region."*
Pancreas	Structured Two-Line ICL	*CAMERA_MOTION: The probe gently fans and tilts in the epigastric region while keeping a mostly fixed window. SCENE_DESCRIPTION: A longitudinal view of the pancreas and surrounding vessels appears in the center of the ultrasound field against the grey abdominal background.*

Table 3: Representative outputs on abdominal and pancreatic ultrasound clips. The Structured Two-Line ICL prompt encourages explicit separation between probe (camera) motion and anatomical scene content, which is desirable for training and guidance applications.

- **Decomposable outputs.** The resulting two-field structure (`CAMERA_MOTION`, `SCENE_DESCRIPTION`) can be consumed by downstream components separately, for example a coverage estimator that uses only camera motion or a report generator that uses only scene content.

6.4 Implications for Ultrasound Training and Guidance

Although this case study is small and qualitative, it illustrates how carefully designed ICL prompts can make general-purpose VLMs more useful for ultrasound applications:

(1) **Probe Handling Feedback.** Clear camera-motion descriptions could help trainees retrospectively review whether they performed adequate sweeps or fanning patterns in a given exam, without retraining a domain-specific model.
(2) **Coverage and Documentation.** Structured captions provide a human-readable trace of which regions were scanned (scene) and how the probe was moved (camera), supporting quality assurance and structured reporting [14].
(3) **Low-Barrier Prototyping.** Because our approach relies only on prompting, clinicians could prototype motion-aware ultrasound feedback tools on existing general-purpose VLMs before committing to specialized datasets or fine-tuning.

These findings suggest that simple, ICL-driven prompt design may offer a practical bridge between research benchmarks like CameraBench and real-world ultrasound training scenarios.

7 Discussion

Our results quantify a fundamental tension in Video Vision-Language Models: the difficulty of maintaining broad semantic understanding while specializing in geometric tasks.

7.1 The Mechanism of the Specialization Trade-off

The data suggests that camera motion and scene understanding compete for finite representational capacity. Fine-tuning on CameraBench improved camera motion scores by 7.5% but degraded scene scores by 4.9%. This implies that the fine-tuning process did not merely add new knowledge but actively reallocated attention toward geometric cues (e.g., optical flow, parallax) at the expense of general semantic object recognition.

7.2 Metric Asymmetry and Evaluation Challenges

A critical finding is the divergence between reference-based and reference-free metrics. While ICL prompts caused BERTScore to drop by 1% to 5%, LLM-as-a-Judge confidence scores remained high (97% - 98%).

- **Length Bias:** Our analysis suggests BERTScore penalizes the structured, concise outputs of our ICL prompts because they are shorter than the detailed reference captions generated by Gemini Pro.
- **Semantic Equivalence:** The high Judge scores confirm that despite the drop in n-gram/embedding overlap, the meaning of the captions remained accurate. This highlights the necessity of using holistic judges when evaluating structured generation tasks.

7.3 Behavioral Rigidity and Instruction Non-Compliance

Beyond capacity limits, we hypothesize that "Behavioral Rigidity" contributes to the failure of ICL. Qualitatively, we observed that the Fine-tuned model often struggled to adhere to the dual-task structure, frequently truncating or omitting the 'SCENE_DESCRIPTION' despite explicit prompting. This suggests that Supervised Fine-Tuning (SFT) on single-task labels may induce a form of catastrophic forgetting regarding instruction following. The model becomes so specialized in outputting camera motion that it ignores secondary prompt constraints, rendering inference-time steering ineffective.

7.4 Implications for Downstream Application

For practitioners in domains like robotics or medical imaging, these findings dictate a clear choice:

- If **geometric precision** is paramount (e.g., visual odometry), use specialized fine-tuned models with default prompts.
- If **holistic understanding** is required (e.g., surgical report generation), generalist models or ensemble approaches are superior.
- **Prompting is not a cure-all** Our results show that prompt engineering cannot reverse the architectural consequences of aggressive fine-tuning. Future work must look toward multi-objective training or modular architectures to solve this duality.

8 Future Work

To overcome the specialization trade-off, future research must move beyond inference-time prompting and address the underlying architectural constraints.

8.1 Architectural Innovations

The most promising direction is to explicitly decouple geometric and semantic reasoning. Architectures that employ separate "expert" modules—one for camera motion (processing optical flow or frame boundaries) and one for scene content (processing semantic embeddings)—could potentially achieve high performance on both tasks without representational competition.

8.2 Joint Training Objectives

Rather than fine-tuning on single-task camera labels, future work should explore multi-objective training frameworks. By optimizing a joint loss function that weights both camera accuracy and scene reconstruction equally, models might learn to allocate capacity more efficiently between these competing dimensions.

8.3 Clinical Validation

While our qualitative ultrasound case study (Section 6) showed promise, rigorous quantitative validation is needed. Future efforts should curate annotated medical video datasets to measure whether the "compliance" of dual-task prompts translates to improved clinical utility in real-world surgical training environments.

9 Conclusion

This work investigates the fundamental asymmetry in video vision-language models' ability to simultaneously understand camera motion and scene content. We identify a critical specialization trade-off: fine-tuning on camera motion data improves motion understanding (+7.5%) but degrades scene understanding (-4.9%). We systematically evaluate three prompting strategies across camera motion and scene understanding tasks, finding that prompts introduce modest BERTScore degradation (0.8%–4.2%) while maintaining or improving LLM-as-judge scores, revealing important metric asymmetries.

9.1 Key Findings

(1) **Specialization Trade-off:** Fine-tuning on camera motion causes representational reallocation that improves target-task performance but measurably harms scene understanding, suggesting that joint optimization of both tasks requires more sophisticated approaches than single-task fine-tuning.
(2) **Metric Complementarity:** BERTScore and LLM-as-judge capture different quality dimensions. BERTScore is sensitive to output formatting (penalizing structured outputs despite semantic equivalence), while judges assess holistic quality robustly. Future video understanding research should employ both metrics.
(3) **Prompting Limitations:** While structured prompts improve judge-perceived quality and help models avoid critical errors, they introduce BERTScore degradation on motion descriptions, suggesting that fundamental architectural or training changes (not prompting) may be necessary to overcome the camera-scene understanding trade-off.

9.2 Future Work

Several directions merit further investigation:

(1) **Joint Training Objectives:** Develop training procedures that optimize for both camera motion and scene understanding, with explicit loss weighting or multi-task learning frameworks to balance competing objectives.
(2) **Broader Model Evaluation:** Extend evaluation to other V-VLM architectures (LLaVA, InternVL, GPT-4o) and parameter scales (3B, 13B, 70B) to assess generalizability of findings.
(3) **Longer-Form Video:** Evaluate on extended cinema, documentaries, and egocentric video to assess performance beyond short clips.
(4) **Architectural Innovations:** Explore architectures that explicitly decouple geometric reasoning (for camera motion) from semantic reasoning (for scene understanding), reducing representational competition [**?**].
(5) **Clinical Validation:** Conduct rigorous quantitative and qualitative evaluation on annotated surgical video datasets with domain expert feedback.

Acknowledgments

We especially thank Junhyeong Cho for initiating this project and providing invaluable guidance on experimental design. Additionally, GPT and Claude were used to aid in the development of the code for this project.

References

[1] Yutao Bai, Zhengyan Zhang, Xiaoyu Li, et al. 2024. Qwen2.5-VL: Enhancing Large Vision-Language Models with Comprehensive Training. *arXiv preprint arXiv:2412.15115* (2024).
[2] Mark S Brown, Ronald H Silverman, and Ming-Song Kuo. 2013. Real-time guidance in ultrasound imaging: A review. *Ultrasound in Medicine & Biology* 39, 10 (2013), 1884–1897.
[3] Rui Chen, Haotian Wang, Jialu Li, and Yuxin Zhao. 2025. Understanding Camera Geometry Errors in Video Vision-Language Models. *arXiv preprint arXiv:2509.18905* (2025).
[4] Google DeepMind. 2025. Gemini 2.5: Scaling Multimodal AI Models. *arXiv preprint arXiv:2507.06261* (2025).
[5] Tianyu Dong, Xiaochen Wang, Elisa Meyerson, et al. 2024. A Survey on In-Context Learning. In *Proceedings of EMNLP*.
[6] Jiawei Gu et al. 2024. A Survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594* (2024).
[7] Sangwoo Lee et al. 2024. VideoLens: A Benchmark for Cinematic and Geometric Understanding. *arXiv preprint arXiv:2407.01516* (2024).
[8] Huawei Li, Jiahui Sun, Rui Zhang, et al. 2025. Contrastive In-Context Learning Improves Constraint Following. *arXiv preprint arXiv:2507.23211* (2025).
[9] Mingxuan Li et al. 2024. Geometry-Aware Evaluation of Video Vision-Language Models. *arXiv preprint arXiv:2402.12289* (2024).
[10] Zongyu Lin, Tianyu Xiao, Jiaqi Xu, Fan Yang, and Jiahui Yu. 2025. CinematicBench: Benchmarking Cinematic Understanding in Vision-Language Models. *arXiv preprint arXiv:2506.21356* (2025).
[11] Zongyu Lin, Jiaqi Xu, Tianyu Xiao, and Jiahui Yu. 2025. CameraBench: A Large-Scale Benchmark for Camera Motion Understanding in Video Models. *arXiv preprint arXiv:2504.15376* (2025).
[12] Hao Liu, Chunyuan Li, Yutong Bai, et al. 2023. Visual Instruction Tuning. *arXiv preprint arXiv:2304.08485* (2023).
[13] Medmastery. 2017. How to perform an ultrasound exam of the pancreas. YouTube. https://www.youtube.com/watch?v=IsJeMgxTFMU.
[14] J Alison Noble and Karim Boukerrou. 2014. Automatic Quality Assessment in Freehand Ultrasound Imaging: A Review. *Medical Image Analysis* 18, 2 (2014), 176–190.
[15] OpenAI. 2024. GPT-4o System Card. *arXiv preprint arXiv:2410.21276* (2024).
[16] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020* (2021).
[17] Yale Radiology and Biomedical Imaging. 2018. How I do it: Ultrasound of the Abdomen. YouTube. https://www.youtube.com/watch?v=i73ovcEL3OI.
[18] Martin G Tolsgaard, Charlotte Ringsted, Eva Dreisler, Anne Klemmensen, Arne Rasmussen, Poul Frederiksen, and Jette L Sorensen. 2014. Ultrasound skill acquisition and assessment: A comprehensive review. *Ultrasound in Obstetrics & Gynecology* 43, 4 (2014), 472–483.
[19] Steve Unger, William McIlhagga, and Michael H Nathanson. 2016. Automatic analysis of probe motion in ultrasound-guided procedures. *International Journal of Computer Assisted Radiology and Surgery* 11, 10 (2016), 1855–1864.
[20] Xuezhi Wang, Juwen Li, Adams Chen, et al. 2023. Large Language Models Can Self-Improve. *arXiv preprint arXiv:2312.04684* (2023).
[21] Jie Wu et al. 2023. Evaluating Temporal Consistency in Video-Language Models. *arXiv preprint arXiv:2307.15818* (2023).
[22] Tianyu Xiao, Chenlin Zhang, Kai Liu, et al. 2024. Role Prompting Improves Factuality and Consistency in Language Models. *arXiv preprint arXiv:2405.02501* (2024).
[23] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.
[24] Wenhao Zhang et al. 2025. Cinematic Precision in Vision-Language Models. *arXiv preprint arXiv:2511.03325* (2025).
[25] Yiming Zhang, Ran He, Li Zhou, et al. 2025. Role-Based Decomposition Improves Multimodal Reasoning in Vision-Language Models. *arXiv preprint arXiv:2509.23501* (2025).
[26] Yifan Zhou, Xu Han, Edward Zhu, et al. 2024. Curriculum In-Context Learning. *arXiv preprint arXiv:2402.10738* (2024).

A Appendix: Code and Reproducibility

All code for this project is available at:

- **GitHub repository:** https://github.com/audreytjokro/dl25-camera-bench

The repository contains:

- Caption generation notebooks for pre-trained and fine-tuned Qwen2.5-VL-7B and Gemini.
- Evaluation notebooks for computing BERTScore and LLM-as-judge metrics and reproducing the main tables and figures.

B Appendix: Prompt Templates

B.1 Unified Dual-Task Prompts (Camera + Scene in One Output)

```
prompt = """You are an expert video analyst. You must describe BOTH:
1. How the CAMERA moves (camera motion).
2. What the SCENE looks like (scene description).

You will ALWAYS output EXACTLY TWO lines in this format:

CAMERA_MOTION: <one short sentence about how the camera moves>
SCENE_DESCRIPTION: <one or two short sentences about what is in the scene>

Important rules:
- CAMERA_MOTION must ONLY talk about how the camera moves in 3D space:
  e.g., pan, tilt, zoom, dolly, truck/slide, roll, orbit, handheld shake.
  You may also mention speed (slow/fast) and steadiness (smooth/shaky).
  Do NOT mention what people or objects are doing in CAMERA_MOTION.
  Do NOT talk about scene content in CAMERA_MOTION.
- SCENE_DESCRIPTION must describe the visible scene:
  main subjects (people/objects), environment (indoor/outdoor, setting),
  notable actions or changes in the scene, and overall context.
- Keep each line concise and natural, like a human-written caption.
- Do NOT mention 'frames', 'video', or technical terms like 'FOV'.
- Do NOT say 'in this video' or 'the video shows'.
- Never output any extra text besides those two lines.

EXAMPLES
========
Example 1:
CAMERA_MOTION: The camera stays almost completely still on a tripod with only a
       slight natural sway.
SCENE_DESCRIPTION: A person stands on a small stage giving a talk in front of
       an audience inside a conference hall.

Example 2:
CAMERA_MOTION: The camera slowly pans from left to right in a smooth,
       continuous motion.
SCENE_DESCRIPTION: A city skyline with tall buildings and a wide river appears
       at sunset, with warm light reflecting on the water.

Example 3:
CAMERA_MOTION: A handheld camera walks forward with small side-to-side shakes.
SCENE_DESCRIPTION: Someone walks down a crowded street lined with food stalls
       and pedestrians.

Now, follow the same format and style for the current video.
Remember:
- Output EXACTLY TWO lines.
- FIRST line must start with 'CAMERA_MOTION:'.
- SECOND line must start with 'SCENE_DESCRIPTION:'.
"""
```

Listing 1: Structured two-line prompt for joint camera and scene description.

```
prompt = """You are a professional video captioner. For each video, you must
       produce ONE short paragraph
that clearly describes BOTH how the camera moves and what the scene looks like.

You will ALWAYS output your answer in this structure:

[CAMERA] <one short sentence, only about how the camera moves.>
[SCENE] <one or two sentences, only about what is in the scene and what is
       happening.>

Guidelines for [CAMERA]:
- Focus ONLY on camera motion: pan, tilt, zoom, dolly, truck/slide, roll,
       orbit, handheld shake, speed, and steadiness.
- Do NOT mention people, objects, or actions in [CAMERA].

Guidelines for [SCENE]:
- Describe the main subjects and the environment (indoor/outdoor, setting).
- Mention the most important action or change in the scene.

General rules:
- Keep [CAMERA] and [SCENE] separate and clearly tagged.
- Do NOT say 'in this video' or 'the video shows'.
- Do NOT mention frames, timestamps, or technical details.
- Output exactly one [CAMERA] line and one [SCENE] line.

EXAMPLES
========
Example 1:
[CAMERA] The camera remains fixed on a tripod with only a barely noticeable
       sway.
[SCENE] A person sits at a table in a small kitchen, talking while a pot
       simmers on the stove.

Example 2:
[CAMERA] The camera slowly pans from left to right in a smooth, controlled
       motion.
[SCENE] A modern open-plan office comes into view with rows of desks and a few
       people working near large windows.

Now produce your answer for the current video using the exact format:
[CAMERA] ...
[SCENE] ...
"""
```

Listing 2: Tagged paragraph prompt for joint camera and scene description.

```
prompt = """You are a video understanding assistant. For each video, first you
       silently think about:
1. The camera's path (direction, speed, and steadiness).
2. The scene: who or what is visible, where they are, and what is happening.

Then, you ONLY output two short sentences with the following format:

CAMERA_MOTION: <ONE sentence, max 25 words, only about camera movement.>
SCENE_DESCRIPTION: <ONE sentence, max 35 words, only about the scene content.>

Detailed instructions:
- CAMERA_MOTION: mention motion types (pan, tilt, zoom, dolly, truck/slide,
       roll, orbit, handheld, static),
  and optionally speed/steadiness. No people or objects here.
- SCENE_DESCRIPTION: briefly describe the main subjects, setting, and key
       action.
- Avoid meta phrases like 'in the video' or 'we see'.

EXAMPLES
========
Example 1:
CAMERA_MOTION: The camera remains fixed on a tripod with only tiny, almost
       imperceptible sway.
SCENE_DESCRIPTION: A person sits at a desk in a home office, typing on a laptop
       with bookshelves behind them.

Example 2:
CAMERA_MOTION: The camera tilts upward while slowly zooming in, keeping the
       center of the frame on the subject.
SCENE_DESCRIPTION: A tall building rises into view against the sky as cars and
       pedestrians move along the street below.

Now, after silently reasoning about the current video, output ONLY:
CAMERA_MOTION: <sentence>
SCENE_DESCRIPTION: <sentence>
"""
```

Listing 3: Chain-of-thought dual-task prompt with two-sentence summary.

```
prompt = """You are an expert video analyst. You must describe BOTH:
1. How the CAMERA moves.
2. What the SCENE contains.

Output exactly TWO lines:
CAMERA_MOTION: <one short sentence only about camera movement>
SCENE_DESCRIPTION: <one or two short sentences only about scene content>

Important rules:
- CAMERA_MOTION: only camera motion (pan, tilt, zoom, dolly, truck/slide, roll,
      orbit, handheld, static).
- SCENE_DESCRIPTION: only visible scene content (subjects, environment, action).
- Do not mention 'video', 'frames', or shot types.

Below are THREE EXAMPLES ordered from easy -> medium -> hard camera motion.

Example 1 (easy):
CAMERA_MOTION: The camera remains almost completely still with a slight natural
      sway.
SCENE_DESCRIPTION: A person stands on a small indoor stage speaking to an
      audience.

Example 2 (medium):
CAMERA_MOTION: The camera slowly pans from left to right in a smooth motion.
SCENE_DESCRIPTION: A city skyline appears with tall buildings and a river at
      sunset.

Example 3 (hard):
CAMERA_MOTION: A handheld camera moves forward with small side-to-side shakes.
SCENE_DESCRIPTION: Someone walks through a crowded outdoor market lined with
      food stalls.

Now describe the current video.

CAMERA_MOTION:
SCENE_DESCRIPTION:
"""
```

Listing 4: Curriculum ICL prompt (easy->medium->hard examples).

```
prompt = """You are an expert video analyst. You must describe BOTH camera
      motion and scene content.

Always output exactly TWO lines:
CAMERA_MOTION: <one short sentence only about camera movement>
SCENE_DESCRIPTION: <one or two short sentences only about scene content>

Important rules:
- CAMERA_MOTION: only camera motion (pan, tilt, zoom, dolly, orbit, handheld,
      static).
- SCENE_DESCRIPTION: only visible scene content.
- Do NOT mix camera and scene between the two lines.

GOOD and BAD examples:

GOOD Example 1:
CAMERA_MOTION: The camera slowly pans from left to right.
SCENE_DESCRIPTION: A city skyline appears with tall buildings and a river at
      sunset.

BAD Example 1 (do NOT follow):
CAMERA_MOTION: A city skyline appears with tall buildings and a river.
SCENE_DESCRIPTION: The camera slowly pans from left to right.

GOOD Example 2:
CAMERA_MOTION: A handheld camera moves forward with small shakes.
SCENE_DESCRIPTION: Someone walks through a crowded outdoor market lined with
      food stalls.

Now ignore the BAD example and describe the current video using only the GOOD
      pattern.

CAMERA_MOTION:
SCENE_DESCRIPTION:
"""
```

Listing 5: Contrastive ICL prompt with GOOD and BAD examples.

```
prompt = """You are TWO coordinated experts analyzing the same video:
(1) CAMERA OPERATOR - describes ONLY how the camera moves.
(2) SCENE OBSERVER - describes ONLY what the scene contains.

Output exactly TWO lines:
CAMERA_OPERATOR: <one sentence only about camera motion>
SCENE_OBSERVER: <one or two sentences only about scene content>

Rules:
- CAMERA_OPERATOR: motion types only (pan, tilt, zoom, dolly, orbit, handheld,
      static).
- SCENE_OBSERVER: subjects, environment, context, and actions only.
- Absolutely no mixing between the two.
- No references to 'video', frames, or shot types.

EXAMPLES
========
Example 1:
CAMERA_OPERATOR: The camera remains almost completely still with a slight
      natural sway.
SCENE_OBSERVER: A person stands on a small indoor stage speaking to an audience.

Example 2:
CAMERA_OPERATOR: The camera slowly pans from left to right.
SCENE_OBSERVER: A city skyline appears with tall buildings and a river at
      sunset.

Now describe the current video.

CAMERA_OPERATOR:
SCENE_OBSERVER:
"""
```

Listing 6: Role-based ICL prompt with CAMERA_OPERATOR and SCENE_OBSERVER.

B.2 Separate Camera-Only and Scene-Only Prompts

```
prompts = {
    "BASE_camera_motion": (
        "Describe the camera motions in this video. Focus only on how the
      camera moves "
        "(pan, tilt, zoom, dolly, truck, orbit, static, handheld)."
    ),
    "BASE_scene_description": (
        "Describe the scene in detail: the main subjects, environment, and
      actions."
    ),
}
```

Listing 7: Base single-task prompts for camera motion and scene description.

```
prompts = {
    "EXPERT_TAXONOMY_camera_motion": """You are a professional cinematographer
      analyzing camera motion using the standard cinematography taxonomy.
Describe all camera motions in this video, covering steadiness
      (static/handheld/stabilized),
translation (dolly, pedestal, truck), rotation (pan, tilt, roll), intrinsic
      changes (zoom),
and object-centric motions (tracking, arc, lead).""",
    "EXPERT_TAXONOMY_scene_description": """You are a professional visual
      analyst. Describe the scene by analyzing subjects (people/objects),
environment (setting, lighting), spatial composition
      (foreground/midground/background),
and temporal changes over the duration of the video.""",
}
```

Listing 8: Expert-taxonomy prompts for camera motion and scene content.

```
prompts = {
    "COT_camera_motion": """Analyze the camera motions step-by-step.
1) Observe frame boundary movement (shifts/rotations).
2) Check parallax between foreground and background.
3) Inspect subject size changes (dolly vs zoom).
4) Check horizon/vertical lines (pan vs truck).
5) Identify tracking if a moving subject stays centered.
Then provide a concise description of all camera motions.""",
    "COT_scene_description": """Analyze the scene step-by-step: identify main
      subjects, their actions, the environment, spatial layout,
and how the scene evolves over time. Then provide a concise description of the
      scene content.""",
}
```

Listing 9: Chain-of-thought prompts for camera-only and scene-only analysis.

```
prompts = {
    "DEF_camera_motion": """Describe camera motions using precise definitions:
- Dolly vs Zoom: dolly creates parallax; zoom does not.
- Pan vs Truck: pan rotates from a fixed point; truck translates sideways.
- Tilt vs Pedestal: tilt pivots; pedestal translates vertically.
Apply these distinctions and cite visual evidence for each motion you
      identify.""",
    "DEF_scene_description": """Describe the scene with attention to
      motion-relevant details: subject motion, depth layers,
reference points (horizon, vertical structures, ground plane), and how the
      scene changes over time.""",
}
```

Listing 10: Definition-grounded prompts with explicit motion and scene distinctions.

Received 15 November 2025