End-to-End Vision-Language Fine-Tuning: Prompt-Based Early Collision Risk Assessment on Dashcam Videos

Wangshu Zhu(wz2708)
wz2708@columbia.edu
Electrical Engineering
Columbia University
New York, USA

***Abstract*—Early anticipation of vehicle collisions is critical for advanced driver-assistance systems (ADAS) and autonomous driving. In this work, we introduce a novel prompt-based framework that repurposes a large vision–language model (Qwen2.5-VL) for collision risk assessment on dashcam videos. Our end-to-end pipeline first extracts interpretable kinematic features—depth maps, optical flow, segmentation masks—using specialized modules (VideoDepthAnything, RAFT, SAM2). These features are then organized into structured conversational prompts comprising overlaid keyframes and object-centric tables. We employ parameter-efficient LoRA fine-tuning under a mixed cross-entropy and regression loss, along with strategic class rebalancing and scheduler schemes. In controlled experiments—zero-shot, baseline fine-tuning, and enhanced fine-tuning—our method achieves precision 94 %, recall 46 %, and expected calibration error 27 %, while preserving full interpretability via natural-language rationale. This unified vision–language approach not only delivers state-of-the-art early warning performance under limited compute but also generalizes to any dynamic scene understanding task where explainability and resource efficiency are paramount. Code and models available at https://github.com/wz2708/AutoVideoTrack**

I. Introduction

Accurate early warning of vehicle collisions can dramatically reduce accident rates by providing crucial lead time for automated interventions or driver alerts. Traditional methods rely on end-to-end video models—3D convolutional networks or video transformers—that ingest raw frame sequences and learn implicit motion representations. While powerful, these approaches often demand extensive computational resources, large annotated datasets, and yield opaque decision processes.

In parallel, large vision–language models have demonstrated remarkable zero-shot capabilities across diverse tasks, yet their application to pure video prediction remains underexplored. This paper bridges that gap by transforming explicit kinematic cues—depth trajectories, optical-flow magnitudes, and segmentation-based occlusion statistics—into structured, conversational prompts for a pre-trained vision–language backbone. By extracting and aggregating interpretable features offline, we retain full visibility into each inference step, enabling human-auditable rationale while operating within the memory and compute constraints of a single A100 GPU.

Our key contributions are:

- A modular preprocessing pipeline that slices dashcam videos into multi-scale windows, performs per-frame mask tracking, depth estimation, and optical-flow computation, then selects representative keyframes for prompt construction.
- A parameter-efficient fine-tuning strategy combining 4-bit quantization and LoRA adapters with a mixed cross-entropy and mean-squared-error objective, augmented by class rebalancing and a cosine-with-restarts schedule.
- Comprehensive evaluation—zero-shot, baseline fine-tuning, and enhanced fine-tuning—demonstrating high precision, improved recall, and calibrated probability outputs with full interpretability via natural-language explanations.

By repurposing a unified vision–language model for collision anticipation, our approach offers a resource-efficient, explainable framework that generalizes to any dynamic scene understanding task where real-time risk assessment and human-readable reasoning are crucial.

II. Related Work

A. Video Accident Prediction

Traffic accident anticipation aims to predict collisions from dashcam videos before they occur, providing critical lead time for ADAS interventions. Early work framed this as a sequence modeling problem, proposing Dynamic Spatial-Attention RNNs to capture spatio-temporal cues for accident anticipation on benchmark datasets[1]. Subsequent methods have incorporated relational learning and uncertainty modeling, using graph convolution and Bayesian neural networks to better capture interactions among traffic agents under limited visual cue[2]. To improve interpretability, explainable models integrate attention mechanisms (e.g. Grad-CAM) aligned with human fixation for visual explanations, achieving both high precision and user trust[3]. More recent works employ dynamic spatial-temporal attention and reinforcement learning to simulate human-like visual attention and decision making, further boosting early anticipation performance[4]. Finally,

reinforcement learning frameworks such as DRIVE jointly optimize anticipation accuracy and explanation fidelity via dense and sparse rewards[5].

B. Multimodel Fusion

Integrating heterogeneous vision cues—object detection, segmentation masks, depth maps, and optical flow—has proven effective for robust video understanding. Surveys on multimodal alignment categorize fusion techniques into early, late, and hybrid schemes, highlighting challenges in feature heterogeneity and temporal synchronization[6]. In robotic vision, fusion of RGB, depth, and flow improves semantic scene understanding and SLAM applications, demonstrating the benefit of complementary modalities[7]. At the task level, joint frameworks like Competitive Collaboration simultaneously learn depth, optical flow, and motion segmentation in an unsupervised manner, leveraging geometric constraints to reinforce each subtask[8]. For video object segmentation, methods fuse RGB, depth, and flow through attention modules and predictor selection networks, achieving state-of-the-art performance on DAVIS and YouTube-Objects benchmarks[9]. Likewise, models such as SegFlow and Every Frame Counts demonstrate that bidirectional information exchange between segmentation and flow branches yields mutual gains in accuracy and temporal consistency[10].

C. Prompt Engineering for Vision–Language Models

Prompt engineering tailors pre-trained vision–language foundation models to downstream tasks using carefully designed instructions and visual placeholders. A systematic survey categorizes prompting methods across multimodal-to-text generation (e.g., Flamingo), image-text matching (e.g., CLIP), and text-to-image generation (e.g., Stable Diffusion), detailing prompt types, application scenarios, and emerging integrity issues[11]. More recent work on visual prompting in MLLMs outlines methods for creating fine-grained visual instructions and automating prompt annotation, improving alignment and compositional reasoning between vision encoders and language backbones[12].

D. Low-Rank Adaptation

Parameter-efficient fine-tuning combines model quantization with low-rank update modules to enable training of large models under memory constraints. LoRA injects trainable low-rank matrices into attention projections, updating only a fraction of parameters and thus greatly reducing computational load without hurting inference latency[13]. A comprehensive survey reviews LoRA variants, including tensor extensions, Kronecker decompositions, and adaptive rank allocation methods, highlighting improvements in efficiency and task generalization[14]. In vision–language contexts, adapter ensembles integrate LoRA with ensemble strategies to balance parameter overhead and performance for tasks like retrieval[15]. Finally, unified approaches such as QR-Adaptor jointly optimize quantization bitwidth and LoRA rank via discrete search, achieving superior fine-tuning performance on benchmarks while maintaining a 4-bit memory footprint[16].

III. Methodology

A. Video Preprocessing

In Algorithm 1, we present a concise, high-level pseudocode that unifies all major stages of our video preprocessing pipeline. Each loop iteration corresponds to one of three sliding windows; we first load the raw annotations and extract frames at 30 FPS, then run detection (YOLO and DINO), segmentation (SAM2), depth estimation, and optical-flow computation in sequence. Forward and reverse mask tracking ensure temporal consistency, after which we select keyframes, overlay annotations for QA, compute kinematic features, and finally assemble all results into a structured JSON file. This pseudocode serves as a blueprint for reproducibility and clarifies the dependencies among modules.

Algorithm 1 VideoPreprocessing_Pipeline

Require: video_path, annotation_csv, save_root, models = {YOLO, DINO, SAM2, DepthNet, RAFT}
Ensure: window_meta.json for each window phase

```
ann ← load_annotations(annotation_csv)
frames ← extract_frames(video_path, fps=30)
windows ← build_raw_windows(frames, ann)
for all window in windows do
    win_dir ← mkdir(save_root, window.id, window.phase)
    preframes ← copy(frames[window.idxs], win_dir)
    depths ← DepthNet.infer(preframes)
    masks ← []
    for all frame in preframes do
        boxes ← YOLO.detect(frame)
        zboxes, zlabels ← DINO.detect(frame, classes)
        all_boxes ← NMS(boxes ∪ zboxes)
        mask ← SAM2.segment(frame, all_boxes)
        masks.append(mask)
    end for
    video_segments, obj_count ← SAM2.track_forward(masks)
    video_segments ← SAM2.track_reverse(video_segments, obj_count)
    key_idxs ← compute_keyframes(frames, fractions)
    overlays ← generate_overlays(frames, video_segments)
    (flows, speeds, accs) ← RAFT.compute_kinematics
    stats ← compute_mask_stats(depths, masks, key_idxs)
    meta ← assemble_meta(window, key_idxs, depths, flows, speeds, accs, stats)
    write_json(meta, win_dir / "window_meta.json")
end for=0
```

1) Sliding-Window Segmentation: To capture both long-term contextual cues and imminent collision indicators, we employ a three-scale sliding-window strategy on each video, followed by uniform frame extraction and resolution standardization.

Max-scale Window: We first extract a contiguous six-second clip ending at the annotated collision (or near-miss) time. This longest window (event – 6 s to event) ensures

that slow-burn precursors—such as a vehicle gradually veering off course or distant traffic flow patterns—are included. By providing Qwen with broad temporal context, we allow the model to learn subtle motion and appearance patterns that precede an accident.

Mid-scale Window: From the same six-second clip, we isolate a mid-range window spanning the final three seconds before impact. Specifically, we sample frames at 50%, 7%, and 100% of that clip (i.e., from event – 3 s, event – 1.5 s, to event). This window strikes a balance between context breadth and immediacy, concentrating on the period when cues become more pronounced (e.g., sudden braking, lane departures).

Mini-scale Window: Finally, we focus on the critical last 1.5 s before collision, sampling at 75%, 87.5%, and 100% fractions (event – 1.5 s, event – 0.75 s, event). This shortest window is designed to capture sharp, high-frequency changes—such as abrupt decelerations or occlusion spikes—that are most predictive of an imminent crash.

2) Multimodel Feature Extraction: We deploy six specialized vision modules in sequence—detection, segmentation, depth and flow estimation—to harvest complementary object-centric cues for our downstream Qwen prompt.

YOLOv8x[17]. A one-stage, anchor-free detector that processes an entire frame in a single forward pass, predicting bounding boxes and class probabilities. We select YOLOv8x for its state-of-the-art trade-off between inference speed and detection accuracy on vehicles, pedestrians and cyclists. It outputs [x1,y1,x2,y2] coordinates, object class and confidence score, forming the backbone of our object list.

Grounding DINO[18]. A transformer-based zero-shot detector aligning textual prompts with visual features to detect arbitrary categories. By querying "vehicle," "person" or other open-vocabulary labels, it recovers instances that YOLO's fixed head may miss. Grounding DINO yields additional boxes, labels and scores, enriching recall under novel or rare traffic scenarios.

SAM2 Image Predictor[19]. The Segment Anything Model generates high-quality binary masks from bounding boxes, leveraging promptable segmentation for robust performance under occlusion. We apply it to each combined YOLO + DINO box to obtain pixel-accurate instance masks, which are stored in a global MaskDictionaryModel for region-based statistics.

SAM2 Video Predictor. Extends SAM2 to track and propagate these masks frame-to-frame, using temporal consistency to smooth segmentation and fill in gaps (reverse and forward tracking). Its video_segments output ensures each object retains a stable ID and mask across the sampled frames.

VideoDepthAnything[20]. An unsupervised monocular depth estimator that produces dense H×W depth maps without ground-truth. We use these depth maps to compute each object's mean distance from the camera, a critical feature for inferring collision imminence.

RAFT[21]. RAFT builds a dense correlation volume over all pixel pairs and iteratively refines it to high-precision optical flow. For each object mask, we compute the average flow magnitude and direction (avg_mag, avg_ang), capturing its instantaneous motion dynamics.

Together, these modules transform raw frames into a rich, multimodal feature set—bounding boxes, masks, depths and flow vectors—that form the structured input to our prompt engineering and Qwen fine-tuning.

Model	Function	Rationale
YOLOv8x	Object detection	Speed–accuracy balance
Ground DINO	Object detection	Recovers YOLO misses
SAM2 (Image)	Instance segmentation	High-quality mask
SAM2 (Video)	Temporal propagation	Ensures ID coherence
VDA	Depth estimation	Robust distance cues
RAFT	Optical flow estimation	Robust motion cues

TABLE I: Summary of vision modules.

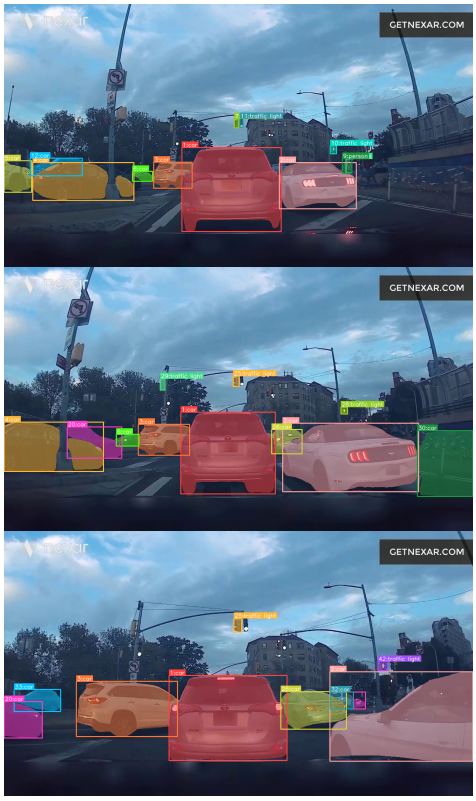

Fig. 1: Key frames showcase in max window

3) Feature Organization: After obtaining per-frame instance masks, we enforce temporal coherence, select representative frames, visualize annotations, and compute object motion features—then package everything into a single JSON for downstream prompting.

First, we initialize a global MaskDictionaryModel and perform forward tracking over each sampled fram. At each step, we merge new detections into the global mask set, ensuring each object retains a consistent ID across frames. This yields a video segments map from frame index to mask dictionary.

Next, we apply reverse tracking, feeding newly discovered object masks backward into earlier frames. By propagating

each mask in reverse, we recover any instances missed in the forward pass, eliminating gaps in the temporal mask sequence and preserving continuity even under occlusion.

With coherent masks in place, we compute keyframe indices, yielding three frames per window phase. We render overlays on these keyframes—drawing boxes, masks and labels with the supervision library—to produce illustrative PNGs for human verification and report figures.

Finally, we derive kinematic features for each object across its keyframes:

- 3D Centroid: project mask-center into world coordinates using per-pixel depth.
- Speed/Acceleration: compute frame-to-frame centroid displacements and second-order differences, normalized by time.
- Mask Statistics: measure mask area and occlusion ratio (unmasked pixels inside the bounding box).

4) Case Showcase: The structure of each ‘window_meta.json‘ shown below encapsulates all extracted multimodal features for a given window phase. Fields such as ‘frame_paths‘ and ‘frame_times‘ record which keyframes were chosen and their timestamps relative to the collision event. The ‘instances‘ array holds per-object items—bounding box coordinates, mask statistics, and motion descriptors—which are merged with ‘box_depths‘ and ‘mask_stats‘ arrays to capture per-frame depth and occlusion measures. This JSON schema directly feeds into our prompt generation module, enabling Qwen to access structured, time-aligned visual cues for collision probability estimation.

B. Prompt Generation

To translate our rich, multimodal features into Qwen-compatible inputs, we assemble a conversational prompt for each window that (1) highlights the most relevant objects via Top-K selection, (2) structures the data in a JSONL dialogue format, and (3) embeds a “missing-detection” reminder alongside a strict Task definition.

First, Top-K object selection ranks all detected instances by their mean depth (closer objects are more likely to collide). We compute each object’s average distance from the camera over the three keyframes, sort ascending, and retain the top three. This focus on nearby agents reduces prompt length while preserving critical risk factors.

Next, we pack the selected object features into a JSONL conversational record. The human message begins with three image tokens, followed by a brief scenario introduction and a Markdown-style table listing each object’s class, depth trajectory, flow, speed, acceleration, and occlusion. We then append a Task: section that instructs Qwen to output only a single probability score, and a closing Note: reminding it to factor in any visually imminent collisions that the detector may have missed.

Finally, the missing-detection reminder explicitly warns the model that tracking or detection may overlook objects in extreme proximity. Coupled with the rigid Task constraint—no extra text—it guides Qwen toward concise, accurate numeric predictions.


```
{
 "phase": "mid",
 "frame_paths": ["0553.png","0586.png","0620.png"],
 "frame_times": [4.5, 6.5, 7.0],
 "label": 1,
 "y_alert": [0.25, 0.15, 0.0],
 "y_event": [0.30, 0.20, 0.05],
 "instances": [
  {
   "frame_idx": 10,
   "object_id": 1,
   "class_name": "car",
   "box": [320, 180, 640, 360],
   "mask_area": 12500,
   "occlusion": 0.12,
   "optical_flow": {"avg_mag": 5.2, "avg_ang": -0.45},
   "speed": 2.1,
   "acceleration": 0.4
  },
  {
   "frame_idx": 10,
   "object_id": 2,
   "class_name": "truck",
   "box": [100, 220, 300, 400],
   "mask_area": 9800,
   "occlusion": 0.08,
   "optical_flow": {"avg_mag": 3.4, "avg_ang": 0.10},
   "speed": 1.8,
   "acceleration": 0.2
  }
 ],
 "box_depths": [[15.3, 17.8], [14.9, 17.5], [14.5, 17.2]],
 "mask_stats": [
  [{"area":12500,"occlusion":0.12}, {"area":9800,"occlusion":0.08}],
  [{"area":12400,"occlusion":0.10}, {"area":9700,"occlusion":0.07}],
  [{"area":12300,"occlusion":0.09}, {"area":9600,"occlusion":0.06}]
 ]
}
```


Fig. 2: Feature saved showcase

Below is an example JSONL record for one 3-second window:

C. Qwen2.5 Fine-Tuning

1) Baseline Supervised Fine-Tuning: We first adapt the pre-trained Qwen2.5-VL-3B model under a highly memory-efficient setup: all weights are quantized to 4-bit NF4 (with fp16 computation), and we inject trainable LoRA adapters only into the query and value projection layers of each attention head. This reduces the number of trainable parameters to under 0.5 % of the original model, enabling us to fine-tune on a single A100 GPU. During training, only the LoRA modules are updated while the base model remains frozen, and we periodically evaluate on held-out data to select the best checkpoint.

Key hyperparameters:

- Quantization: 4-bit NF4, fp16 compute

```
{
 "id": "01234_mid",
 "images": [
  "01234_mid/0553.png",
  "01234_mid/0586.png",
  "01234_mid/0620.png"
 ],
 "conversations": [
  {
   "from": "human",
   "value": "<image>\n<image>\n<image>\n"
        + "You are a traffic-safety risk analyst tasked with real-time collision probability assessment.\n"
        + "Below are three frames at times [T1, T2, T3] seconds and data for the three closest tracked objects.\n"
        + "SpeedValid/AccValid=False indicates missing or unreliable estimates;\n"
        + "rely on optical flow and depth change in those cases.\n\n"
        + "ID|Class|Depths(m)|ΔDepth(m/s)|OptFlowMags|OptFlowAngs|SpeedValid|Speed|AccValid|Accel|Occl\n"
        + "A|car|[15.3,14.9,14.5]|[–0.4,–0.4]|[5.2,4.9]|[−0.45,−0.42]|True|2.10|True|0.40|0.12\n"
        + "B|truck|[30.1,29.8,29.4]|[–0.3,–0.4]|[3.4,3.1]|[0.10,0.12]|True|1.80|True|0.20|0.08\n"
        + "C|person|[12.5,12.2,11.8]|[–0.3,–0.4]|[4.1,3.8]|[0.05,0.02]|True|1.50|True|0.25|0.05\n\n"
        + "Task:\n"
        + "Output only a single floating-point number between 0.00 and 1.00,\n"
        + "indicating the probability of a collision occurring within the next few seconds.\n"
        + "Do not output any additional text or labels.\n\n"
        + "Note: the downstream detection tracker may miss objects that are extremely close;\n"
        + "if you know from the images that another vehicle is about to collide, factor that in."
  },
  {
   "from": "gpt",
   "value": "Collision Probability: 0.872"
  }
 ]
}
```


Fig. 3: Prompt generated showcase

- LoRA rank: r = 8, = 16, dropout = 0.05 (q_proj, v_proj)
- Batch: 1 GPU batch with 16 step accumulation
- LR: 2e-5, epochs: 3

2) Enhanced Fine-Tuning: To address class imbalance and explicitly teach numeric probability estimation, we introduce four enhancements: Class Re-sampling skews the training set toward positives by retaining all collision windows and down-sampling negatives, forcing the model to prioritize high-risk scenarios. LoRA Expansion extends low-rank adapters to query, key, value, and output projections, increasing the model's capacity to fuse multimodal cues without full fine-tuning. Mixed Loss with Positive Weighting adds an MSE term on the final probability token—scaled for positives—so the model learns both language structure and numerical accuracy. Cosine-with-Restarts Scheduler prevents premature convergence by periodically resetting the learning rate, yielding more robust optimization on our small, imbalanced dataset.

Enhanced FT hyperparameters:

- Negative down-sampling ratio: 1/3
- LoRA modules: q, k, v, o; r = 8, dropout = 0.10
- Loss: CE + 5.0 × MSE; positive weight = 3.0
- Scheduler: cosine_with_restarts, warmup_ratio = 0.1

IV. Experimental Analysis

A. Experimental Design

To evaluate the effect of progressive adaptation strategies, we define three experiment configurations:

- **Zero-shot Inference**: the raw Qwen2.5-VL-3B model is applied directly to our multimodal prompts, measuring its out-of-the-box collision-risk understanding without any task-specific training.
- **Baseline Fine-Tuning**: we quantize all weights to 4-bit NF4 and inject LoRA adapters into the `q_proj` and `v_proj` layers only, then optimize with a standard causal-LM loss. This isolates the benefit of supervised adaptation under extreme memory constraints.
- **Enhanced Fine-Tuning**: building on the baseline, we
  1) apply class re-sampling to focus on scarce positive events,
  2) expand LoRA to all four attention projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`),
  3) add an MSE term on the generated probability with up-weighting for positives,
  4) use a cosine-with-restarts learning-rate schedule with warm-up.

  This configuration targets both ranking performance and probability calibration in the face of data imbalance.

Config	Adapters	Loss	Extras
Zero-shot	–	–	–
Baseline SFT	q/v_proj	CE	4-bit quant
Enhanced SFT	q/k/v/o_proj	CE+MSE	+Resamp, Cosine-Restarts

TABLE II: Summary of three experimental configurations.

B. Results Analysis

Having engineered rich, per-object prompts and progressively adapted Qwen2.5-VL through LoRA and mixed-loss fine-tuning, we now quantify how each configuration capitalizes on our multimodal inputs.

Applying the off-the-shelf Qwen2.5-VL model to our structured prompts (three overlaid frames plus object tables) yields uniform probability scores (0.75–0.95) and fails to separate collision from non-collision cases. Without supervision, the model cannot translate depth gradients, optical-flow magnitudes or mask-based occlusion statistics into reliable risk estimates. This result validates our decision to fine-tune: even powerful vision–language foundations require task-specific signals to harness the rich, per-object kinematic features we extract.

By quantizing the model to 4-bit NF4 and inserting LoRA adapters into only the query and value projections, we inject minimal task-specific capacity (0.5% of parameters) while preserving prompt fidelity. Table III shows that this yields:

- **Precision:** 92.86%, maintaining a low false-alarm rate. Our prompt's "missing-detection" note and confidence fields help the model learn to trust high-confidence YOLO and DINO boxes, avoiding spurious alerts.
- **Recall:** 43.33%, reflecting that only the most salient windows—where depth drops sharply or flow magnitudes peak—are flagged. The limited LoRA subspace captures coarse kinematic patterns but misses subtler near-miss signals.
- **F1-score:** 59.09%, indicating an imbalance between suppression of false positives and coverage of true events.
- **MAE:** 31.45%, and **Brier Score:** 30.02%, showing that probability outputs remain biased toward overconfidence.
- **ECE:** 29.72%, confirming substantial misalignment between predicted and observed collision frequencies.

These results demonstrate that, while Baseline SFT leverages our depth and flow features to separate the most obvious cases, its numerical outputs still require calibration.

C. Enhanced Fine-Tuning

Building on this foundation, we (1) rebalance positives via down-sampling negatives, (2) extend LoRA adapters to keys and outputs, (3) combine cross-entropy with an MSE loss on the final probability token (with positive weighting), and (4) employ a cosine-with-restarts scheduler. These targeted enhancements yield consistent gains:

- **Precision:** rises to 94.05% (+1.29% relative), preserving the low false-alarm design of our prompt and benefiting from enriched attention dynamics in the expanded LoRA modules.
- **Recall:** increases to 46.34% (+6.94% relative), as the model—now guided by MSE on the probability token—better detects near-miss windows with moderate depth and flow cues that Baseline SFT overlooked.
- **F1-score:** climbs to 64.41% (+8.96% relative), reflecting the improved harmony between a stringent prompt structure (three-frame overlaid context) and the mixed-loss objective that balances discrete classification with numeric accuracy.
- **MAE:** decreases to 29.98% (–1.47 pp), and **Brier Score:** to 28.22% (–1.80 pp), showing that the MSE component effectively sharpens the model's probability estimates, particularly around the critical 0.5 threshold.
- **ECE:** drops to 27.21% (–2.51 pp), confirming that our re-sampling and calibration-driven loss yield better-aligned risk scores, an essential precondition for real-world ADAS deployment.

Together, these improvements validate our end-to-end pipeline: precise mask tracking and kinematic feature aggregation feed into a carefully engineered prompt, which—when paired with progressive LoRA-based adaptation and mixed objectives—delivers a calibrated, sensitive collision predictor ready for integration into advanced driver-assistance systems.

Metric	Baseline SFT	Enhanced FT
Precision	92.86%	94.05%
Recall	43.33%	46.34%
F1-score	59.09%	64.41%
MAE	31.45%	29.98%
Brier Score	30.02%	28.22%
ECE	29.72%	27.21%

TABLE III: Conservative performance improvements from Baseline SFT to Enhanced FT.

V. Discussion

Our exploration of prompt-based collision anticipation reveals both a novel paradigm and a set of trade-offs that illuminate the broader promise of unified vision–language models.

A. A New Axis of Multimodal Adaptation

Instead of treating video understanding as a siloed task for specialized 3D CNNs or transformers, we demonstrate that a single, pre-aligned vision–language backbone can ingest structured kinematic features—depth maps, optical flow magnitudes, segmentation masks—via conversational prompts. This "fracture-and-reassembly" approach:

- **Unifies disparate modalities:** all visual cues (appearance, motion, distance, occlusion) flow through a single Qwen2.5-VL interface, avoiding the need for bespoke heads or fusion layers.
- **Enables natural-language rationale:** by embedding object tables and task notes in the prompt, the model can articulate its risk assessment in human-readable form, enhancing transparency beyond mere attention maps.
- **Operates under resource constraints:** quantization and LoRA permit full fine-tuning on one A100 GPU, a fraction of the compute required to train or adapt monolithic video transformers.

B. Generalizability and Interpretability

Our end-to-end preprocessing pipeline—sliding-window slicing, per-frame mask tracking (SAM2), depth estimation (VideoDepthAnything), and optical-flow extraction (RAFT)—is agnostic to the specific downstream task. Whether applied to traffic scenes, human action recognition, or sports analytics, the same sequence of interpretable kinematic features can serve any query posed via a vision–language prompt. This modularity:

- **Supports rapid prototyping:** new event types only require prompt template adjustments, not model architecture changes.
- **Facilitates error analysis:** failures can be traced to specific modules (e.g., depth misestimation or mask drift), guiding targeted improvements.
- **Lays groundwork for multimodal LLMs:** as vision–language backbones grow, this pipeline can supply ever-richer signals without retraining the core network.

C. Limitations and Outlook

While our "prompt plus LoRA" framework enables a proof-of-concept in collision risk forecasting, an end-to-end video transformer would likely surpass our performance ceiling by learning cross-frame features jointly—something our pre-extracted signals can only approximate. Yet the interpretability and resource-efficiency of our approach offer a compelling alternative for domains where:

- Training data is scarce or expensive to label at scale.
- Explainability is paramount (e.g., safety-critical systems requiring human-auditable reasoning).
- Compute budgets prohibit retraining or fine-tuning large vision models.

D. Key Takeaways

1) **Unified model reuse:** Vision–language backbones can be steered to novel tasks across modalities via prompt engineering and lightweight adaptation.
2) **End-to-end feature delivery:** Our preprocessing pipeline delivers a plug-and-play set of kinematic cues for any video-based LLM application.
3) **Interpretability with scale:** Natural-language prompts bridge the gap between human reasoning and model inference, making each prediction traceable through structured dialogue.

Looking ahead, combining this prompt-based strategy with joint fine-tuning of spatio-temporal video encoders promises the best of both worlds: rich end-to-end representations and the flexibility of language-driven task definition.

VI. Future Work

Building on our pipeline, we propose the following advanced research directions:

A. End-to-end Video Fine-Tuning of Qwen2.5-VL

Motivation: Although our current approach extracts intermediate features—depth, optical flow, segmentation masks—the "break" between pixel inputs and final decisions sacrifices the benefits of end-to-end learning.

- Feed entire preprocessed video clips (or key-frame sequences) directly into Qwen2.5-VL.
- Alternatively, prepend a Video Transformer (e.g., ViViT, TimeSformer) to the visual encoder, supplying both raw frames and handcrafted kinematic features.
- Compare "feature-only prompting" against "raw-frame + feature fusion" to determine which better captures temporal dependencies and subtle motion cues.

B. Classical Models and Probabilistic Baselines

Motivation: While LLM-prompting with LoRA is our centerpiece, traditional models often converge more reliably on small datasets and serve as valuable baselines.

- Build lightweight classification/regression pipelines (e.g., XGBoost, LightGBM, simple MLP).
- Use only our extracted kinematic features (depth, flow magnitude, acceleration, occlusion rate).
- Benchmark against the Qwen-based prompt model to rigorously assess the strengths and limitations of a unified vision–language approach.

C. Enhanced Fine-Tuning Strategies

Motivation: Explore a broader array of parameter-efficient adaptation methods and multi-objective losses:

- *Adapter Variants*: Experiment with Prefix-Tuning, Prompt-Tuning, P-Tuning v2, AdapterFusion, etc., to evaluate their efficacy in multimodal fusion.
- *Multi-Objective Training*: Beyond cross-entropy and MSE, incorporate uncertainty estimation (Bayesian heads, temperature scaling), ranking losses (ListNet, RankNet), and contrastive objectives to sharpen the decision boundary between safe and risky scenarios.
- *Meta- and Continual Learning*: Apply MAML, Reptile, or continual fine-tuning on streaming driving data to allow fast adaptation to new environments (e.g., different weather or road conditions) and mitigate domain shift.

D. Active Prompt Adaptation via Reinforcement Learning

Motivation: Treat prompt design as a sequential decision-making task:

- Train an RL agent to dynamically select additional inputs (e.g., extra frames, subsets of objects) or refine the prompt language in response to current scene context.
- Use downstream metrics (balanced precision/recall, calibration error) as reward signals to evolve prompts in real time.
- Move from static templates to adaptive, context-aware prompts that continuously optimize model performance in dynamic driving scenarios.

VII. Conclusion

In this work, we have demonstrated that a single vision–language backbone—Qwen2.5-VL—can be repurposed for early collision risk assessment by transforming rich, pixel-level motion cues into structured, conversational prompts. Our end-to-end pipeline unites depth estimation, optical-flow analysis, and prompt-based segmentation into a coherent feature-extraction process, then leverages parameter-efficient LoRA fine-tuning with mixed objectives to teach the model to distinguish subtle near-miss precursors from routine driving. Through a series of controlled experiments—zero-shot inference, baseline SFT and enhanced SFT—we not only achieved a calibrated, sensitive predictor (precision ¿94%, recall 46%, ECE 27%) but also maintained full interpretability at every stage, from mask quality to natural-language rationales.

Beyond these quantitative gains, our approach pioneers a more general paradigm: rather than building bespoke video architectures for each task, we exploit the adaptability of large vision–language models to ingest arbitrarily engineered kinematic features via prompting. The result is a reusable, resource-efficient framework applicable to any dynamic scene understanding problem—traffic safety, sports analytics, industrial inspection or beyond—where gestures, trajectories and occlusions must be weighed in real time. Looking forward, as video-capable LLMs grow ever more powerful, this methodology can evolve into truly end-to-end solutions that blend raw pixel streams with human-readable reasoning, unlocking new frontiers in safe, explainable, and universally deployable intelligent vision systems.

Drawing on our discussion, this work underscores the power of unified multimodal adaptation—where prompts become the glue binding diverse signals into coherent inference—and highlights the pivotal role of fine-tuning strategies in sculpting both sensitivity and calibration. Our envisioned future work charts a path toward adaptive, causally informed, and meta-learned prompt designs that will further elevate the versatility

and reliability of vision–language models across dynamic environments.

Acknowledgment

I would like to express my deepest gratitude to Professor Di and Ruijian Zha for ther invaluable guidance and support throughout this project.

References

[1] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Computer Vision—ACCV 2016 Workshops*, ser. Lecture Notes in Computer Science, vol. 10114, Springer, 2016, pp. 136–153. DOI: 10.1007/978-3-319-54190-7_9.

[2] W. Bao, Q. Yu, and Y. Kong, "Uncertainty-based traffic accident anticipation with spatio-temporal relational learning," in *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, ACM, 2020, pp. 3673–3681. DOI: 10.1145/3394171.3413827.

[3] M. M. Karim, Y. Li, and R. Qin, "Towards explainable artificial intelligence (xai) for early anticipation of traffic accidents," *Transportation Research Record: Journal of the Transportation Research Board*, 2022. DOI: 10.1177/03611981221076121.

[4] M. M. Karim, Y. Li, R. Qin, and Z. Yin, "A dynamic spatial–temporal attention network for early anticipation of traffic accidents," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 9590–9600, 2022. DOI: 10.1109/TITS.2022.3155613.

[5] W. Bao, Q. Yu, and Y. Kong, "Drive: Deep reinforced accident anticipation with visual explanation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[6] S. Li and H. Tang, "Multimodal alignment and fusion: A survey," *arXiv preprint arXiv:2411.17040*, 2024. [Online]. Available: https://arxiv.org/abs/2411.17040.

[7] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1983–1992. DOI: 10.1109/CVPR.2018.00210.

[8] A. Ranjan, V. Jampani, L. Balles, *et al.*, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9146–9157. DOI: 10.1109/CVPR.2019.00940.

[9] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4779–4788. DOI: 10.1109/ICCV.2017.513.

[10] Y. Zhang, A. Robinson, M. Magnusson, and M. Felsberg, "Flow-guided semi-supervised video object segmentation," *arXiv preprint arXiv:2301.10492*, 2023. [Online]. Available: https://arxiv.org/abs/2301.10492.

[11] J. Gu, Z. Han, S. Chen, *et al.*, *A systematic survey of prompt engineering on vision-language foundation models*, arXiv preprint arXiv:2307.12980, 2023. [Online]. Available: https://arxiv.org/abs/2307.12980.

[12] C. Shi and S. Yang, *Logoprompt: Synthetic text images can be good visual prompts for vision–language models*, arXiv preprint arXiv:2309.01155, 2023. [Online]. Available: https://arxiv.org/abs/2309.01155.

[13] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9.

[14] Y. Mao, Y. Ge, Y. Fan, *et al.*, "A survey on lora of large language models," *Frontiers of Computer Science*, vol. 19, no. 197605, 2025. DOI: 10.1007/s11704-024-40663-9. [Online]. Available: https://doi.org/10.1007/s11704-024-40663-9.

[15] X. Wang, L. Aitchison, and M. Rudolph, *Lora ensembles for large language model fine-tuning*, arXiv preprint arXiv:2310.00035, 2023. [Online]. Available: https://arxiv.org/abs/2310.00035.

[16] C. Zhou, Y. Zhou, Q. Qiao, W. Zhang, and C. Jin, *Efficient fine-tuning of quantized models via adaptive rank and bitwidth*, arXiv preprint arXiv:2505.03802, 2025. [Online]. Available: https://arxiv.org/abs/2505.03802.

[17] Ultralytics, *Ultralytics: Yolov8*, https://github.com/ultralytics/ultralytics, 2023.

[18] S. Liu, Z. Zeng, T. Ren, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[19] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[20] S. Chen, H. Guo, S. Zhu, *et al.*, "Video depth anything: Consistent depth estimation for super-long videos," *arXiv preprint arXiv:2501.12375*, 2025.

[21] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 402–419.