2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

KITE: Keyframe-Indexed Tokenized Evidence for VLM-Based Robot Failure Analysis

Mehdi Hosseinzadeh, King Hang Wong, and Feras Dayoub

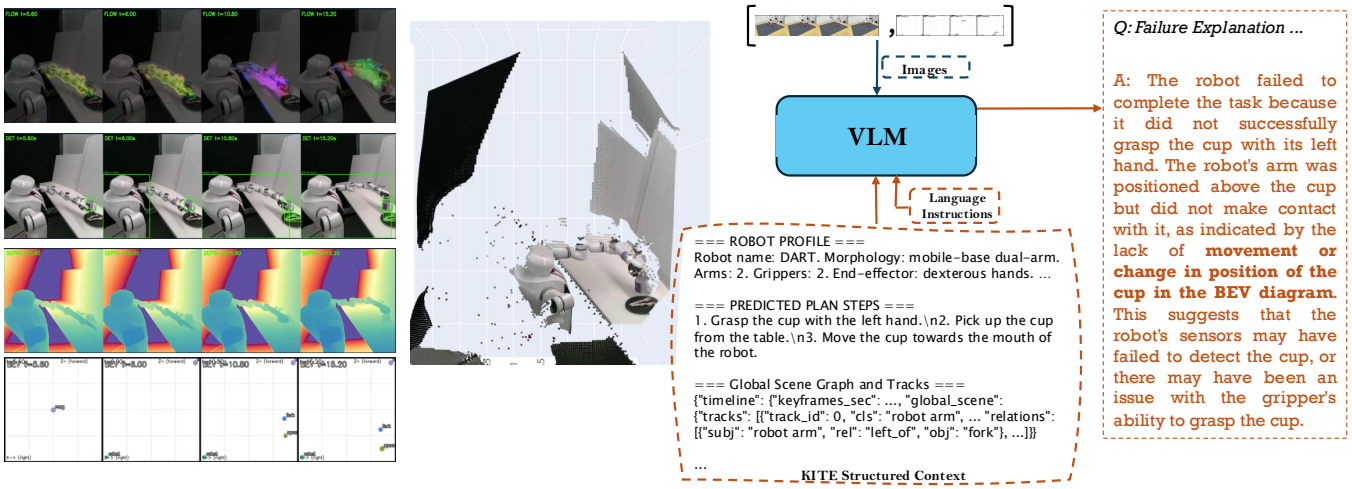

Fig. 1: **Failure explanation in real-world settings with KITE.** Example sequence from the Dual-Arm Robot (DART) in the lab. Left (top to bottom): optical flow estimates used for keyframe selection; RGB keyframe with object detection overlays; single-view depth estimates; and a pseudo-BEV schematic (circles with radius $\propto$ confidence; X/Z axes; timestamp). Notably, in its failure explanation, the VLM references the BEV diagram to infer that the cup's position remains unchanged.

***Abstract*— We present KITE, a training-free, keyframe-anchored, layout-grounded front-end that converts long robot-execution videos into compact, interpretable tokenized evidence for vision-language models (VLMs). KITE distills each trajectory into a small set of motion-salient keyframes with open-vocabulary detections and pairs each keyframe with a schematic bird's-eye-view (BEV) representation that encodes relative object layout, axes, timestamps, and detection confidence. These visual cues are serialized with robot-profile and scene-context tokens into a unified prompt, allowing the same front-end to support failure detection, identification, localization, explanation, and correction with an off-the-shelf VLM. On the RoboFAC benchmark, KITE with Qwen2.5-VL substantially improves over vanilla Qwen2.5-VL in the training-free setting, with especially large gains on simulation failure detection, identification, and localization, while remaining competitive with a RoboFAC-tuned baseline. A small QLoRA fine-tune further improves explanation and correction quality. We also report qualitative results on real dual-arm robots, demonstrating the practical applicability of KITE as a structured and interpretable front-end for robot failure analysis. Project page: `https://m80hz.github.io/kite/`**

I. Introduction

Robots executing long-horizon manipulation in the wild still fail in mundane ways: a gripper approaches a mug off-axis and slides off, a handle is contacted too late relative to the arm's motion, or a bimanual handover misaligns in space and time. Explaining such errors requires combining *where* (layout, contact, relative pose), *when* (the moment things go wrong), and *what* (intent and subgoals). However, off-the-shelf VLMs struggle with raw video: subtle cues are often buried among dense visual details, and most models have limited temporal memory, making it difficult to reason over sequences spanning multiple frames.

Prior work attempts to mitigate this challenge by summarizing experiences for an LLM [1], training failure-specific VLMs [2], or introducing QA benchmarks [3]. However, these approaches typically require additional and costly fine-tuning or training on high-parameter models. To the best of our knowledge, there is no *training-free* representation that renders the key spatiotemporal facts immediately legible to an off-the-shelf generalist VLM.

To this end, we propose Keyframe-Indexed Tokenized Evidence (KITE), a compact and interpretable front-end that converts a long execution into a small set of motion-salient keyframes, each paired with a *pseudo-BEV* schematic of object layout and a minimal scene/interaction summary (object relations and a contact proxy). This temporally indexed storyboard is fed to a VLM with a prompt, enabling training-free failure detection, identification, localization, explanation, and correction. We detail KITE in Section III.

In summary, the contributions of this paper are as follows:

Authors are with the Australian Institute for Machine Learning (AIML), Adelaide University, Australia.

- A training-free, keyframe-anchored, layout-grounded representation that serializes robot state into VLM-readable evidence;
- Pseudo-BEV schematics that externalize spatial layout, timestamps, and detection confidence;
- A keyframe-indexed failure localization method, in contrast to approaches that localize failures only at coarse plan steps.

On RoboFAC, KITE with a general-purpose VLM outperforms vanilla Qwen2.5-VL and is competitive with a RoboFAC-tuned baseline, while a small QLoRA fine-tune further improves explanations and corrections. We also demonstrate qualitative effectiveness of our framework by deploying on rollout episodes from two real dual-arm robots (RealMan Dual Arm Compound Robot [4] and ALOHA-2 [5]).

II. Related Work

Foundation Models for Robotics. Recent advances in Large-Language-Models (LLMs) [6]–[9] and Vision-Language-Models (VLMs) [10]–[14] have catalyzed their integration into robotics across planning, control, and interaction domains. Numerous works leverage pretrained LLMs as high-level planners for robots, using natural language understanding and commonsense to decompose tasks and guide actions. For example, [15] pioneered grounding an LLM in robotic affordances for instruction-following, and subsequent systems [16]–[24] have combined language and perception in embodied models that reason over visual inputs to plan robot behavior [25]–[27]. LLM-driven policies have been applied to navigation and mobile manipulation tasks, including domestic assistive robots that follow instructional/guiding prompts to tidy environments or perform user requests. Such LLM-based planners and embodied agents have demonstrated flexible task generalization and improved semantic understanding in novel scenarios.

Robotic Failure Analysis and Summarization. Early works in explainable robotics investigated how to communicate execution errors to users through scripted or learned explanations. [28] generated natural language explanations of failure cases to aid human recovery, while [29] introduced narration of robot experiences, albeit in limited domains. Recent approaches have turned to LLM reasoning to generalize failure diagnosis. The [1] framework is a notable example. They combine multisensory logs into a hierarchical memory and prompt an LLM to infer about failures, producing explanations that can inform recovery plans. Other works exploit LLMs to assist failure recovery via in-context reasoning [30], [31], or train an LLM-based model to summarize and answer questions about robot's past experience [32].

Failure detection and explanation also have a long history in HRI [33]–[36] and in task and motion planning based approaches [37]. The widespread integration of LLMs and VLMs into manipulation pipelines—whether to specify reward functions or synthesize trajectories in zero-shot regimes [38], [39] has renewed interest in recognizing task failures [2], [40]–[43]. Recent methods often use pretrained VLMs/LLMs as success classifiers [43]–[45], and some instruction-tune VLMs for failure detection [46]. Training bespoke models can be resource-intensive and may sacrifice the open-world flexibility of foundation models. To bridge the gap, recent studies have explored structured scene representations as a means to inject inductive bias for reasoning without extensive retraining. Spatial and schematic abstractions can help models reason about key object relationships and dynamics in a failure. For example, [47] grounds an LLM's planning decisions in an explicit 3D scene graph of the environment, and [48] uses a bird's-eye view (BEV) map as an interface for a VLM to perform driving scenario reasoning and Q&A. Diagrammatic and sketch-based representations have likewise been shown to improve visual reasoning [49].

Inspired by these approaches, KITE introduces an interpretable front end that transforms raw execution videos into sequences of keyframe-anchored tokens. This tokenized schematic summary is then provided to a pretrained VLM, enabling analysis and explanation of robot failures without task-specific fine-tuning. By leveraging a structured representation of the trajectory, KITE preserves essential spatial and temporal context for failure diagnosis while remaining training-free and inherently interpretable.

III. Method: KITE Front-End

KITE is a *training-free*, model-agnostic front-end that converts a long robot-execution video into a compact bundle of motion-salient *keyframes*, their *pseudo-BEV* schematics, and a serialized state summary (*tokenized evidence*). The representation is *layout-grounded* and *temporally indexed*, making it legible to general-purpose VLMs. An overview appears in Figure 2.

A. Preliminaries and Notation

Let a video be $\mathscr{V} = \{(I_t, t)\}_{t=1}^{T}$ with $I_t \in \mathbb{R}^{H\times W\times 3}$ an RGB frame and t its timestamp. We select a small budget of M keyframes

$$K = \{(I_{t_k}, t_k, i_k)\}_{k=1}^{M}$$

where i_k is the frame index. For each keyframe k we compute: (i) an open-vocabulary detection set $O_k = \{(b_j, c_j, s_j)\}$ with box b_j, class c_j, confidence s_j; (ii) a relative depth map D_k (single-view, up to scale); and (iii) a contact proxy label $\gamma_k \in \{\textsc{Gain}, \textsc{Loss}, \textsc{Stable}\}$ between the robot hand/gripper and its nearest object (if visible).

B. Keyframe Selection

We operate under a small budget M and prioritize motion-salient frames. To detect salient events, we compute dense optical flow between consecutive frames and score each frame by its average flow magnitude. Keyframes are selected as local peaks in this score using temporal non-maximum suppression. If fewer than M salient frames are identified, we supplement them with uniformly spaced frames to preserve contextual coverage. The selector is modular and can be replaced with entropy-based or learned policies; our ablations compare peak-based and uniform selection strategies.

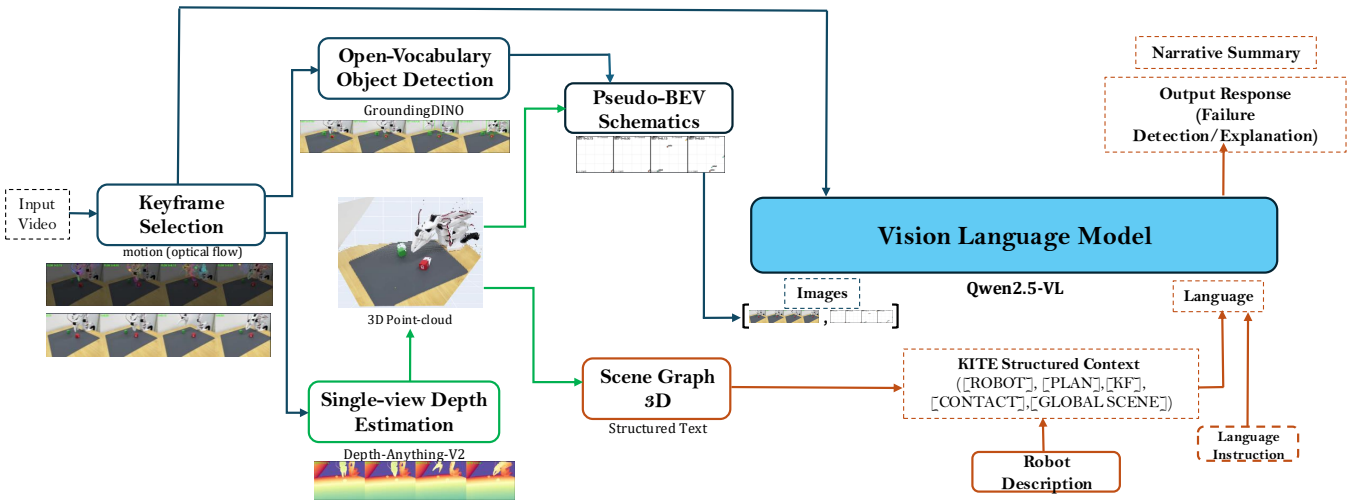

Fig. 2: **Overview of KITE.** The proposed pipeline takes a raw video and distills it into a small set of salient keyframes, identified using motion-based peaks. For each keyframe, we run open-vocabulary detection to localize the robot and surrounding objects, and render a pseudo-BEV schematic that depicts the scene layout with simple, interpretable symbols. These visual elements are paired with a structured context and form a compact, interpretable front-end for prompting a vision-language model. The model can then answer diverse failure analysis QA tasks, as well as generate grounded explanations and final narratives. The sequence illustrated here comes from the real-world subset of the RoboFAC dataset [3].

C. Per-Keyframe Perception

Open-Vocabulary Detection (OVD): We run an OVD module (e.g., GroundingDINO [50]) to detect objects of interest and robot arms/grippers. Detections are temporally linked across keyframes into short tracks (instance IDs) and timestamps t_k are rendered as numbered overlays on the RGBs.

Single-View Depth Estimation: We estimate *relative* depth per keyframe (e.g., Depth-Anything-V2 [51]) and associate depth statistics with each detection, enabling coarse 3D ordering needed for relations and the pseudo-BEV layout.

Contact Proxy: To provide a simple yet informative signal about the relative state of the robot and the object of interest—common in manipulation tasks—we design a coarse contact-proxy token for each keyframe, when a robot hand or gripper and an object are detected with high confidence. To capture interaction state, we compute the nearest-center distance d_k between the gripper and its closest object, along with the bounding-box IoU, IoU_k, for that pair at keyframe k. With thresholds $\tau_{\mathrm{IoU}}, \tau_d > 0$.

$$\gamma_k = \begin{cases} \textsc{Gain}, & \mathrm{IoU}_{k+1} - \mathrm{IoU}_k \geq \tau_{\mathrm{IoU}} \quad , \quad d_{k+1} - d_k \leq -\tau_d, \\ \textsc{Loss}, & \mathrm{IoU}_k - \mathrm{IoU}_{k+1} \geq \tau_{\mathrm{IoU}} \quad , \quad d_{k+1} - d_k \geq \tau_d, \\ \textsc{Stable}, & \text{otherwise.} \end{cases}$$

This coarse token is robust and cheap to compute.

D. 3D Scene Graph

For each keyframe we build a local scene graph G_k whose nodes are detections in O_k with 3D centroids approximated from relative depth and camera geometry. We encode pairwise relations $\{\textsc{left\_of}, \textsc{above}, \textsc{in\_front}\}$ using sign and magnitude of centroid offsets (with small tolerance thresholds). Local graphs $\{G_k\}_{k=1}^{M}$ are aggregated into a global graph by maintaining instance tracks; with a fixed camera, aggregation reduces to tracking moving objects and merging consistent IDs. The global graph (nodes, tracks, relations) is serialized as structured JSON for the KITE context.

E. Robot Description

We include a concise robot profile (morphology: #arms, #grippers, end-effector types; sensors; workspace; salient constraints), enabling the VLM to condition explanations and proposed corrections on embodiment and environment.

F. Pseudo-BEV Schematic (Layout Prior)

Photorealistic reconstructions are costly and misaligned with what current VLMs parse well. We therefore render a *schematic*, non-metric top-down **pseudo-BEV** per keyframe that externalizes spatial layout while preserving identity consistency across modalities:

- fixed axes (X right, Z forward) with arrows;
- one circle per tracked object (circle radius $\propto s_j$), class label c_j, and the same instance ID used in RGB;
- overlaid timestamp t_k and keyframe index.

Pseudo-BEVs are *layout-aware cues* (not metrically accurate); they accompany RGB keyframes to provide a clean, VLM-legible depiction of relative placement and timing.

G. KITE: Keyframe-Indexed Tokenized Evidence

We serialize a compact, interpretable context prefix that serves as a *single front-end across all QA tasks*. Let $\mathcal{T}$ denote

the KITE context string:

$$\mathscr{T} = \underbrace{\texttt{[ROBOT]}\ \text{short description}}_{\text{morphology, gripper, workspace}} \parallel \underbrace{\texttt{[PLAN]}\ \text{high-level plan}}_{\text{numbered natural-language steps}} \parallel \underbrace{\texttt{[KF}\ i_k\ \texttt{@}\ t_k\texttt{]}}_{\text{timestamped keyframe tags}} \parallel \underbrace{\texttt{[CONTACT]}\ \gamma_k}_{\text{GAIN/LOSS/STABLE}} \parallel \underbrace{\texttt{[GLOBAL\_SCENE]}\ \text{tracks \& relations}}_{\text{IDs consistent with RGB/pseudo-BEV}}$$

Plan steps can be optionally generated by a VLM from the same inputs; otherwise the `[PLAN]` field is left empty.

H. Prompting and Failure Localization

For each question, we provide a small image bundle (RGB keyframe overlays + corresponding pseudo-BEVs) and prepend $\mathscr{T}$ to the text prompt. We include one instruction clarifying that the BEV is a *schematic, not to scale*, and should be used for *relative layout*. For *failure localization* (frame-level), we request *strict JSON*:

```
{"candidates":[{"frame_num": INT,
"confidence": FLOAT}, ...]}
```

with up to three candidates, confidence $\in [0,1]$. A simple parser extracts the top candidate; visualization then aligns subsequent analysis to that frame. (Parsing details and the exact prompt are provided in the supplement.)

I. Narrative Summary

Given $\mathscr{T}$ and a storyboard montage (all keyframes and pseudo-BEVs), we prompt the VLM for a concise, causal narrative that *references keyframe IDs and timestamps* and proposes one high-level and one low-level correction. All perception is per-keyframe; overall cost scales linearly in M and is independent of video length once M is fixed.

IV. Experiments

We evaluate KITE on a large-scale failure analysis dataset, RoboFAC [3], with both quantitative and qualitative analyses, compare against their finetuned model, and Qwen2.5-VL [52] alone as our strong baselines, and report ablations on design choices. We provide ablations for pseudo-BEV, and keyframe selection strategy. We also include qualitative results on sequences from our lab's robots, a RealMan dual-arm compound robot [4] (DART) and ALOHA-2 Stationary [5] (dual arm) to illustrate generalization beyond the benchmark. RoboFAC [3] only contains single-arm tasks.

A. Datasets and Tasks

RoboFAC: RoboFAC [3] is a large QA-style benchmark for robotic failure analysis with both simulation and real-world sequences. It has over 60K QA pairs from the simulated environment as the training set, and includes 10K simulated QA pairs and 8K QA pairs from real-world data for test set. It includes multiple question types: *Task identification (TI)*, *Task planning (TP)*, *Failure detection (FD)*, *Failure identification (FI)*, *Failure locating (FL)*, *Failure explanation (FE)*, *High-level correction (HL)*, *Low-level correction (LL)*. We follow the dataset's official splits and evaluation protocols where applicable, and we compute textual metrics for non-MCQ questions using standard NLP measures.

TABLE I: Performance of multi-modal baseline models on the RoboFAC Benchmark [3]. Success rate for MCQ questions is reported (higher is better) for both simulation and real-world tasks.† denotes the models that are finetuned on RoboFAC benchmark.

Model	Simulation			Real-world		
	FD	FI	FL	FD	FI	FL
Gemini-2.0	0.48	<u>0.27</u>	**0.75**	0.60	0.11	0.18
GPT-4o	<u>0.64</u>	0.21	<u>0.71</u>	**0.96**	<u>0.43</u>	0.52
Qwen2.5-VL-3B	0.38	0.04	0.51	0.04	0.03	0.07
Qwen2.5-VL-7B	0.52	0.26	0.22	0.83	0.38	<u>0.72</u>
KITE + Qwen2.5-VL-7B	**0.88**	**0.44**	0.55	<u>0.84</u>	**0.43**	**0.74**
RoboFAC-7B†	0.91	0.63	**0.94**	0.80	0.56	0.71
KITE+Qwen2.5-7B+QLoRA†	**0.93**	**0.69**	0.92	**0.89**	**0.58**	**0.77**

DART and ALOHA-2 (in-lab).: We additionally test KITE qualitatively on in-lab sequences from DART and ALOHA robots. These sequences are zero-shot with respect to our method and serve to demonstrate generalization of the proposed method. We include a scene with bi-manual handover as well.

B. Backbones and Baselines

We adopt Qwen2.5-VL [52] as the main backbone due to its strong general vision–language capabilities and open-source availability. We additionally report results for Gemini-2.0 [14], GPT-4o [10], vanilla Qwen2.5-VL-3B and 7B models (without KITE; RGB keyframes only), the RoboFAC model finetuned on the RoboFAC benchmark [3], our training-free KITE + Qwen2.5-VL, and our KITE + Qwen2.5-VL further enhanced with QLoRA.

C. Metrics

For multiple-choice format questions (MCQ) (*FD*, *FI*, and *FL*), we report success rate. For descriptive tasks (*TI*, *FE*, *HL*, and *LL*), following [2], we compute ROUGE-L F1 score (assesses the quality of generated text in tasks like summarization by measuring the similarity between the candidate text and a reference text, focusing on the Longest Common Subsequence of words), and also semantic Sentence-BERT cosine similarity between the embeddings of reference sentence and predicted sentence.

D. Simulation and Real-world Results

We report the performance (success rate) of our proposed method against baselines across different MCQ dimensions for both the simulation and real-world tasks in Table I. We also present results on descriptive question tasks, measured by ROUGE-L and S-BERT cosine similarity, in Table II for both simulation and real-world settings.

Training-free KITE surpasses the Qwen2.5-VL-7B baseline substantially in simulation—achieving gains of +36% on FD, +18% on FI, and +33% on FL—and remains competitive on real-world tasks. Applying QLoRA further improves performance across all tasks.

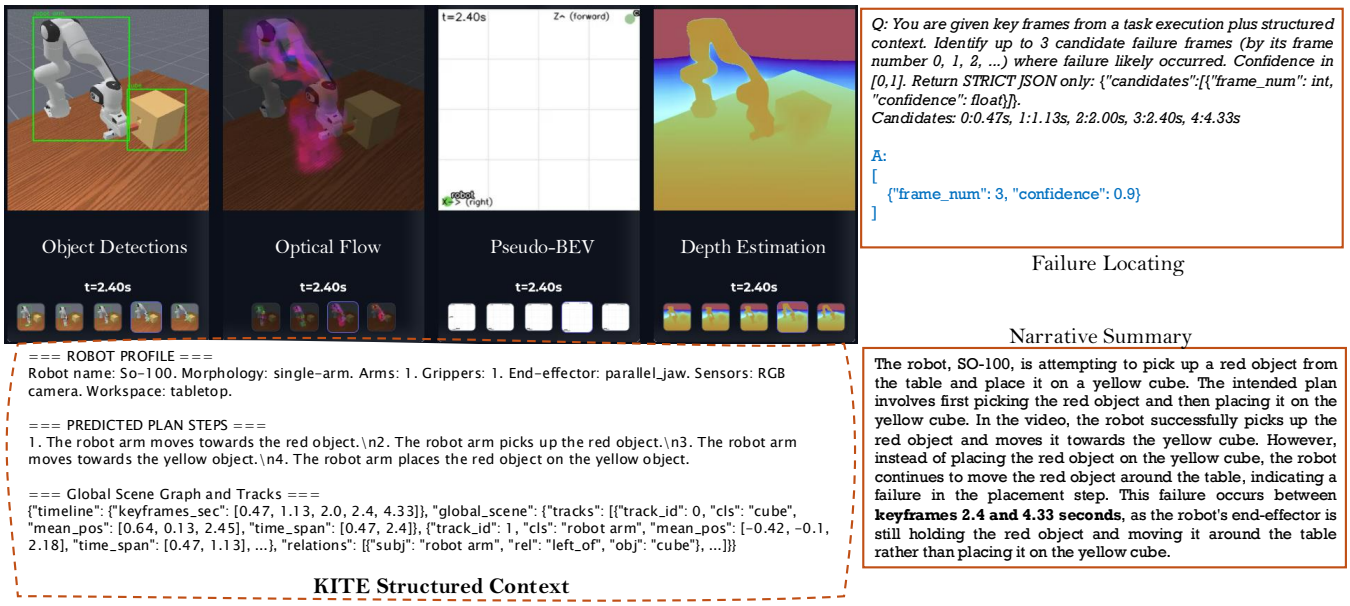

Fig. 3: **Qualitative results in simulation (RoboFAC dataset).** Each panel shows: RGB keyframe with object-detection overlays; optical-flow estimates; pseudo-BEV schematic (consistent object IDs; circle radius $\propto$ confidence; timestamp); and single-view depth estimates, all for the corresponding keyframes. We also illustrate a short structured-context excerpt, KITE's response to a failure-localization query, and a final narrative summary.

TABLE II: Performance of multi-modal baseline models on the RoboFAC Benchmark [3]. ROUGE-L and SBERT cosine similarity metrics (higher is better) are reported for free-language reasoning tasks for both simulation and real-world tasks. † denotes the models that are finetuned on RoboFAC benchmark.

Model	Sim (ROUGE-L)				Sim (SBERT Cosine)				Real (ROUGE-L)				Real (SBERT Cosine)			
	TI	FE	HL	LL	TI	FE	HL	LL	TI	FE	HL	LL	TI	FE	HL	LL
Qwen2.5-VL-7B	0.206	0.194	0.230	0.157	0.546	0.448	0.683	0.657	0.264	0.233	0.219	0.197	0.689	0.786	0.792	0.785
KITE + Qwen2.5-VL-7B	0.295	0.248	0.241	0.190	0.680	0.829	0.798	0.779	0.300	0.252	0.223	0.232	0.696	0.832	0.791	0.804
RoboFAC-7B†	0.323	0.299	0.301	0.245	0.701	0.842	0.808	0.794	0.337	0.361	0.228	0.305	0.722	0.856	0.798	0.813
KITE+Qwen2.5-7B+QLoRA†	0.326	0.314	0.302	0.296	0.698	0.845	0.806	0.803	0.338	0.365	0.229	0.313	0.724	0.860	0.798	0.815

TABLE III: Ablation study of our method. The ↓ indicates the feature is removed. Success rate for MCQ and ROUGE-L metric for other question dimensions are reported for real-world tasks.

Config	success rate			ROUGE-L			
	FD	FI	FL	TI	FE	HL	LL
Full (KITE)	0.84	0.43	0.74	0.300	0.252	0.223	0.232
↓ pseudo-BEV	0.81	0.37	0.70	0.302	0.202	0.221	0.228
uniform keyframe	0.69	0.33	0.56	0.298	0.189	0.217	0.190

E. Ablations

We isolate the contribution of the pseudo-BEV component and study the impact of different keyframe selection strategies–motion-based (default) vs. uniform–on real-world tasks, as shown in Table III. As observed, removing the pseudo-BEV leads to a reduction in FE by 0.05 (ROUGE-L score). Using uniform keyframe selection degrades performance more significantly, particularly for questions involving detection and locating failures (FD, FL, FE).

F. Qualitative Analyses

For a sequence from the "PegInsertionSide" simulation task in the RoboFAC dataset, we illustrate all selected keyframes, object detections, optical-flow estimations, rendered pseudo-BEVs, and depth estimations in Figure 3. As shown in the narrative summary, our method produces a grounded explanation, serving as a robotic introspection narrator.

We also present the same intermediate representations and final outputs for real-world sequences recorded in our lab: DART in Figure 1, and ALOHA-2 in Figure 4. As shown in Figure 4, during the handover of an object ("fork") from the left arm to the right arm, the object is dropped. This failure is captured in the generated explanation, highlighting the impact of incorporating robot descriptions into our structured context.

G. Implementation Details

a) Keyframes: We use up to $M = 8$ keyframes per video, selected primarily from motion-salient peaks detected via optical flow. If fewer than M salient frames are identified, we supplement them with uniformly spaced frames to maintain

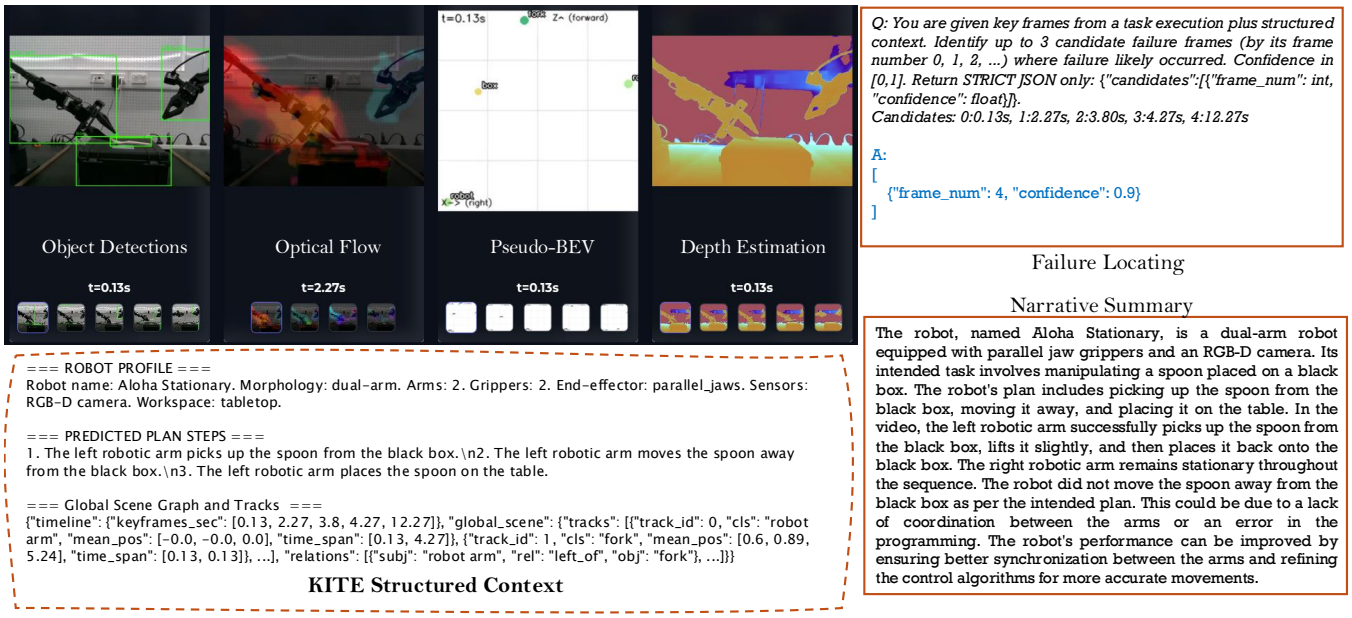

Fig. 4: **Qualitative results in real-world (ALOHA-2).** Each panel shows: RGB keyframe with object-detection overlays; optical-flow estimates; pseudo-BEV schematic (consistent object IDs; circle radius $\propto$ confidence; timestamp); and single-view depth estimates, all for the corresponding keyframes. We also illustrate a short structured-context excerpt, KITE's response to a failure-localization query, and a final narrative summary.

contextual coverage. All images are normalized to a resolution of 512×512 for VLM inputs.

b) Optical Flow, OVD, Depth, and Contact: We compute the per-keyframe mean optical-flow magnitude to propose keyframes by flow, using a simple dense optical flow estimation [53]. We employ GroundingDINO [50] (with Swin-T backbone) for open-vocabulary object detection, and we cap detections at max 5 detections per keyframe. For single-view depth estimation, we use Depth-Anything-V2-Large [51] and to reduce the impact of extreme large depth outliers, we mask out the depth estimations outside the 0.8 quantile bound. Contact proxy uses IoU and nearest-center trends across adjacent keyframes.

c) Contact Proxy: For contact-proxy generation, we adopt a simple yet effective design that captures coarse interaction state (see Section IV). We set the thresholds to $\tau_{\text{IoU}} = 0.1$ for bounding-box overlap and $\tau_d = 0.15$ for nearest-center distance.

d) Pseudo-BEV: For rendering pseudo-BEV schematics, we use a 256×256 white canvas; X/Z axes rendered at the corners; semantic dots for tracked objects (projected X,Z from camera frame); and circle dot radius proportional to the detection confidence score s_j (clipped to $[r_{\min} = 3, r_{\max} = 10]$ pixels); object class labels; and timestamp at top-left (OCR-friendly).

e) VLM Calls: For each QA, we pass a list of $2 \times M$ images: RGB keyframes, and their corresponding pseudo-BEVs. The text prompt comprises the KITE prefix, an instruction for BEV schematics ("... *BEV image is a schematic top-down layout (not to scale); use it for relative spatial layout* ...") and finally the question.

f) QLoRA: To further study the capabilities of our proposed context, we also finetuned the VLM with QLoRA [54] on QA tasks to compare the performance gain against the finetuned baseline. We finetune with rank 8, 4-bit quantization, 1 epoch, with unfrozen LLM backbone and merger parameters with a learning rate of 1×10^{-5}. We use one NVIDIA A6000 GPU for both training and evaluation.

V. Limitations and Future Work

KITE makes a few simplifying assumptions. It relies on open-vocabulary detection and monocular relative depth, which can struggle with small, occluded, or reflective objects, and its contact proxy captures coarse interaction trends rather than precise force events. The pseudo-BEV is intentionally non-metric and flattens vertical structure, so fine 3D relations are not preserved. Keyframe selection based on motion saliency may miss low-motion or very brief failures, and identity tracking can occasionally switch in cluttered scenes. Finally, results depend on the general-purpose VLM backend, which may produce variable reasoning quality. Despite these factors, KITE consistently improves over vanilla VLM baselines, and future work can integrate stronger perception modules, multi-view layout cues, or adaptive keyframe policies to address these challenges.

VI. Conclusion

We introduced KITE, a training-free, keyframe-anchored, pseudo-BEV–grounded front end that converts videos into compact, interpretable tokenized evidence (object-overlaid keyframe RGBs, pseudo-BEVs, robot descriptions, and serialized scene graphs with object tracks) for VLMs. KITE consistently helps a general VLM *locate*, *identify*, and *explain*

failures on RoboFAC, surpasses other baselines (without fine-tuning), and enables concise, causal narratives that explicitly cite evidence frames. KITE is model-agnostic, and qualitative results on real-world sequences from DART and ALOHA-2 suggest that the approach extends beyond a single dataset. In future work, we will explore lightweight perception improvements for small or occluded objects and investigate interactive grounding (e.g., asking targeted follow-ups in ambiguous cases).

REFERENCES

[1] Z. Liu, A. Bahety, and S. Song, "Reflect: Summarizing robot experiences for failure explanation and correction," *arXiv preprint arXiv:2306.15724*, 2023.

[2] J. Duan, W. Pumacay, N. Kumar, Y. R. Wang, S. Tian, W. Yuan, R. Krishna, D. Fox, A. Mandlekar, and Y. Guo, "Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation," in *ICLR*, 2025.

[3] W. Lu, M. Ye, Z. Ye, R. Tao, S. Yang, and B. Zhao, "Robofac: A comprehensive framework for robotic failure analysis and correction," 2025. [Online]. Available: https://arxiv.org/abs/2505.12224

[4] RealMan Robotics, "Compound robot - realman robotics," https://www.realman-robotics.com/compound-robot, 2024, accessed: 2024-09-07.

[5] A. . Team, J. Aldaco, T. Armstrong, R. Baruch, J. Bingham, S. Chan, K. Draper, D. Dwibedi, C. Finn, P. Florence, S. Goodrich, W. Gramlich, T. Hage, A. Herzog, J. Hoech, T. Nguyen, I. Storz, B. Tabanpour, L. Takayama, J. Tompson, A. Wahid, T. Wahrburg, S. Xu, S. Yaroshenko, K. Zakka, and T. Z. Zhao, "Aloha 2: An enhanced low-cost hardware for bimanual teleoperation," 2024.

[6] F. Zeng, W. Gan, Y. Wang, N. Liu, and P. S. Yu, "Large language models for robotics: A survey," *arXiv preprint arXiv:2311.07226*, 2023.

[7] C. Zhang, J. Chen, J. Li, Y. Peng, and Z. Mao, "Large language models for human–robot interaction: A review," *Biomimetic Intelligence and Robotics*, vol. 3, no. 4, p. 100131, 2023.

[8] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[9] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[10] OpenAI, "Hello gpt-4o," May 2024. [Online]. Available: https://openai.com/index/hello-gpt-4o

[11] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023.

[12] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, "Llava-next: Improved reasoning, ocr, and world knowledge," January 2024. [Online]. Available: https://llava-vl.github.io/blog/2024-01-30-llava-next

[13] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[14] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[15] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*. PMLR, 2023, pp. 287–318.

[16] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 230–244, 2022.

[17] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, Z. Zhao, *et al.*, "Toward general-purpose robots via foundation models: A survey and meta-analysis," *arXiv preprint arXiv:2312.08782*, 2023.

[18] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman, *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *arXiv preprint arXiv:2312.07843*, 2023.

[19] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[20] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[21] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[22] M. Crosby, M. Rovatsos, and R. Petrick, "Automated agent decomposition for classical planning," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 23, 2013, pp. 46–54.

[23] B. Xu, Z. Peng, B. Lei, S. Mukherjee, Y. Liu, and D. Xu, "Rewoo: Decoupling reasoning from observations for efficient augmented language models," *arXiv preprint arXiv:2305.18323*, 2023.

[24] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[25] F. Liu, K. Fang, P. Abbeel, and S. Levine, "Moka: Open-vocabulary robotic manipulation through mark-based visual prompting," *arXiv preprint arXiv:2403.03174*, 2024.

[26] H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao, "Copa: General robotic manipulation through spatial constraints of parts with foundation models," *arXiv preprint arXiv:2403.08248*, 2024.

[27] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.

[28] D. Das, S. Banerjee, and S. Chernova, "Explainable ai for robot failures: Generating explanations that improve user assistance in fault recovery," in *Proceedings of the 2021 ACM/IEEE international conference on human-robot interaction*, 2021, pp. 351–360.

[29] S. Rosenthal, S. P. Selvaraj, and M. M. Veloso, "Verbalization: Narration of autonomous robot experience." in *IJCAI*, vol. 16, 2016, pp. 862–868.

[30] S. S. Raman, V. Cohen, I. Idrees, E. Rosen, R. Mooney, S. Tellex, and D. Paulius, "Cape: Corrective actions from precondition errors using large language models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 070–14 077.

[31] Z. Wang, B. Liang, V. Dhat, Z. Brumbaugh, N. Walker, R. Krishna, and M. Cakmak, "I can tell what i am doing: Toward real-world natural language grounding of robot experiences," *arXiv preprint arXiv:2411.12960*, 2024.

[32] C. DeChant, I. Akinola, and D. Bauer, "Learning to summarize and answer questions about a virtual robot's past actions," *Autonomous robots*, vol. 47, no. 8, pp. 1103–1118, 2023.

[33] S. Ye, G. Neville, M. Schrum, M. Gombolay, S. Chernova, and A. Howard, "Human trust after robot mistakes: Study of the effects of different forms of robot communication," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2019, pp. 1–7.

[34] P. Khanna, E. Yadollahi, M. Björkman, I. Leite, and C. Smith, "User study exploring the role of explanation of failures by robots in human robot collaboration tasks," *arXiv preprint arXiv:2303.16010*, 2023.

[35] J. Arkin, D. Park, S. Roy, M. R. Walter, N. Roy, T. M. Howard, and R. Paul, "Multimodal estimation and communication of latent semantic knowledge for robust execution of robot instructions," *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1279–1304, 2020.

[36] A. Bucker, L. Figueredo, S. Haddadin, A. Kapoor, S. Ma, S. Vemprala, and R. Bonatti, "Latte: Language trajectory transformer," *arXiv preprint arXiv:2208.02918*, 2022.

[37] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," in *Proceedings of the international conference on automated planning and scheduling*, vol. 30, 2020, pp. 440–448.

[38] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[39] Y. J. Ma, W. Liang, H.-J. Wang, S. Wang, Y. Zhu, L. Fan, O. Bastani, and D. Jayaraman, "Dreureka: Language model guided sim-to-real transfer," *arXiv preprint arXiv:2406.01967*, 2024.

[40] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer:

Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[41] J. Duan, W. Yuan, W. Pumacay, Y. R. Wang, K. Ehsani, D. Fox, and R. Krishna, "Manipulate-anything: Automating real-world robots using vision-language models," *arXiv preprint arXiv:2406.18915*, 2024.

[42] M. Skreta, Z. Zhou, J. L. Yuan, K. Darvish, A. Aspuru-Guzik, and A. Garg, "Replan: Robotic replanning with perception and language models," *arXiv preprint arXiv:2401.04157*, 2024.

[43] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Conference on Robot Learning*. PMLR, 2023, pp. 3766–3777.

[44] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," *arXiv preprint arXiv:2210.00030*, 2022.

[45] L. Wang, Y. Ling, Z. Yuan, M. Shridhar, C. Bao, Y. Qin, B. Wang, H. Xu, and X. Wang, "Gensim: Generating robotic simulation tasks via large language models," *arXiv preprint arXiv:2310.01361*, 2023.

[46] Y. Du, K. Konyushkova, M. Denil, A. Raju, J. Landon, F. Hill, N. de Freitas, and S. Cabi, "Vision-language models as success detectors," *arXiv preprint arXiv:2303.07280*, 2023.

[47] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning," *arXiv preprint arXiv:2307.06135*, 2023.

[48] T. Choudhary, V. Dewangan, S. Chandhok, S. Priyadarshan, A. Jain, A. K. Singh, S. Srivastava, K. M. Jatavallabhula, and K. M. Krishna, "Talk2bev: Language-enhanced bird's-eye view maps for autonomous driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 345–16 352.

[49] C. Deng, S. Chen, D. Chen, Y. He, and Q. Wu, "Sketch, ground, and refine: Top-down dense video captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 234–243.

[50] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[51] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *arXiv:2406.09414*, 2024.

[52] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, "Qwen2.5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[53] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian conference on Image analysis*. Springer, 2003, pp. 363–370.

[54] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *arXiv preprint arXiv:2305.14314*, 2023.