Amortized Meta-Reasoning: Exploring, Verifying, and Compiling Agentic Grounding Traces for Kilometre-Scale Video QA

Venkata Sai Nikhil Thodupunuri and Ravi Vayuvegula

t.v.s.nikhil123@gmail.com, vayuvegula@gmail.com

Abstract. The KilometerVision track of the Perception Test challenge poses metric- and geometry-grounded questions – line-of-sight *distance*, *compass* facing, turn-order *directions*, *map-trace* shape matching, loop *revisit*, and *landmark* identity – over kilometre-scale, first-person walking-tour videos. Even a single large language model (LLM) given a metric 3D reconstruction of the walk answers only 48.74% of them, because these quantities – metres, bearings, and route topology – must be measured and cross-checked per question rather than read off in one forward pass. We cast spatial grounding as **amortized meta-reasoning**. On a small *explore* split (the validation set plus ∼10% of the test set) an autonomous agent searches, per question, over complementary grounding strategies – object-anchor metric scale, scene-text OCR with map matching, reconstruction-trace shape, and direct frame inspection – and *verifies* each answer by reconciling independent estimates, leaving an interpretable trace. A lightweight *compile* step then distils these traces into a single reusable policy: one agentic loop steered by a learned tool-routing skill and a set of recurring-strategy skills, with an LLM verifier at the commit gate. This policy answers the remaining ∼90% of the test set with no per-question search. The system attains **90.4**% accuracy – first place, and 15.2 points above the runner-up (75.25%) – while keeping every decision auditable.

1 Introduction

KilometerVision asks six kinds of spatial question about long, first-person walking-tour videos: line-of-sight *distance*, *compass* facing, turn-order *directions*, *map-trace* shape matching, loop *revisit*, and *landmark* identity. Each answer depends on physical quantities – metres, bearings, and route topology – that are not explicit in the forward pass of a vision–language model (VLM), so a single such call is unreliable (Sec. 4).

Recovering these quantities reliably requires per-question search: reconstructing geometry, selecting the right measurement, and cross-checking it. On our curated metadata this search follows a clear scaling trend – accuracy rises steadily with the search budget, but cost rises with it too – so running full search at inference for every question is accurate yet expensive. Our central idea is to *pay for search once and reuse it*, which we frame as **amortized meta-reasoning**.

The *explore* stage meta-reasons – for a small set of questions it decides which logical decomposition and which tools recover a question's answer-bearing attributes – and the *compile* stage amortizes those decisions into a single reusable policy. No model weights are updated, which distinguishes this from gradient-based meta-learning. Concretely, we explore at high effort only on the validation set plus ∼10% of the test set; extending the search to the whole test set would multiply token cost for little gain, whereas a ∼10% budget is enough to capture the reasoning patterns the compiled policy needs.

Contributions. (i) An *agentic explorer* that grounds each question by searching over grounding strategies and tools with explicit verification, producing an interpretable per-question trace. (ii) A *compile* step that induces, from these traces, a single policy – one agentic loop parameterised by a learned tool-routing skill, recurring-strategy skills, and an LLM verifier – that amortizes search across the test set. (iii) A first-place challenge result (90.4%), together with a trace-level analysis of 986 autonomous sessions that explains *how* the system reasons.

2 Related Work

A large body of video-QA benchmarks probes spatio-temporal understanding. These benchmarks [1, 3, 7–13, 15, 17, 21–23, 28–30, 32] typically emphasize recognizing objects, identifying actions, and understanding events that are directly observable within individual frames or short clips – whereas KilometerVision demands metric and geometric quantities that no single frame reveals. Long egocentric video QA stresses spatio-temporal reasoning beyond frozen VLMs [14,16]. Agentic tool-use loops – ReAct [31], Toolformer [19], and plan-and-solve prompting [26] – interleave reasoning with tool calls to gather evidence before answering, and chain-of-thought prompting [27] exposes the intermediate inference. Geometric grounding builds on structure-from-motion [20] and, more recently, feed-forward reconstruction (VGGT [24], VGGT-Ω [25]); because such reconstruction is only metric up to scale, absolute scale must be anchored to known-size objects or geo-references. Our compile step (Sec. 3.2) is a form of trace-based workflow induction: distilling repeated agent behaviour into a fixed policy trades one-time exploration for cheap, repeatable inference. Conceptually, the explore stage is *meta-reasoning* – deciding what computation to run under bounded resources [18] – and explore-then-compile is a form of *amortized inference* [4]; both are orthogonal to gradient-based meta-learning [2], which instead adapts parameters across tasks.

3 Method

Our system has two parts: an offline *metric grounding substrate* that turns each raw video into aligned, metric geometry (Sec. 3.1), and an *explore–compile–deploy* procedure that first searches for per-question grounding strategies and then amortizes them into one policy (Sec. 3.2).

3.1 Metric Grounding Substrate

Offline and once per video, we build a reusable substrate that places the camera path and the scene in a single metric coordinate frame.

3D Reconstruction. We construct the metric world model in four steps.

1. *Windowed reconstruction.* We sample each video uniformly (∼5 fps) and reconstruct it in overlapping clips with the feed-forward model VGGT-Ω [25] – 128 frames per clip with a 64-frame overlap – each clip recovered only up to an unknown similarity transform in its own local frame.
2. *Seam stitching.* Adjacent clips share their 64 overlapping frames, whose per-pixel 3D points give direct 3D–3D correspondences. Per seam we estimate a similarity transform (rotation, translation, scale) with a RANSAC Umeyama fit and chain the seams into one global coordinate frame for the whole video.
3. *Adaptive re-reconstruction.* Seam quality anti-correlates with camera rotation, so we score per-frame drift using turn rate and reconstruction confidence, re-segment the clip boundaries accordingly, and re-reconstruct the neighbourhood of any weak seam in a single larger pass (up to 512 frames, again with 64-frame overlap), dissolving the seam.
4. *Stitch-and-scale refinement.* Absolute scale, which single-view reconstruction cannot recover, is anchored to known-size objects that recur across videos – chiefly *person*, *licence plate*, and *bicycle*. Gemini 3.1 Pro [5] is used only to detect and box each object; its true size m_k comes from a fixed prior table (person height 1.70 m, licence-plate width 0.52 m, bicycle-wheel diameter 0.68 m), while its extent e_k is *measured* from the reconstruction. Holding the chained rotations R_i fixed, we solve jointly for the per-clip scales s_i and translations t_i by minimising seam-matching error and object-size error,

$$\min_{\{s_i,\,t_i\}} \sum_{\text{seams}} w_g \left\| s_i R_i p + t_i - s_j R_j q - t_j \right\|^2 + \sum_{\text{anchors}} w_a \left(s_k\, e_k - m_k \right)^2, \quad (1)$$

where clips i and j meet at a seam with matched 3D points $p \leftrightarrow q$; the triple (s_i, R_i, t_i) is the scale, (fixed) rotation, and translation that map clip i into the global frame; k indexes detected anchor objects, with measured extent e_k, known true size m_k, and s_k the scale of the clip in which anchor k lies; and w_g, w_a weight the seam (geometry) and anchor terms. The objective is linear in the scales and translations (s, t) and solved in closed form with Huber reweighting: the seam term fixes relative scale between clips, the anchor term the absolute metric level.

Dense Captioning. We densely caption the sampled frames with *Gemini 3.1 Pro*, producing per-frame descriptions of landmarks, storefronts, signage, and other salient objects. These captions enrich the per-question context digest with a semantic index over the walk that complements the geometric indices and supplies candidates for landmark identification, scale anchoring, and geo-grounding.

Indexing. For every frame i we record, in the shared metric frame, the camera pose (3D position and orientation), the viewing direction, the reconstructed 3D point cloud, and scene-text OCR of on-scene signage. These indices form an aligned chain, frame → 3D position (`trajectory`$[i]$) → top-down 2D trace (`path_2d`$[i]$) → object-anchor metric scale, exposed to the agent through a per-question bundle and a compact context digest.

Geo-Grounding. As an optional refinement, we register the reconstructed route to a real road network. This succeeds only for videos with recognisable landmarks and clean OCR that together triangulate a location, so it is applied as an override rather than a default. A geo-grounding agent ingests the scale-normalised VGGT-Ω route and the per-frame OCR and coordinates two proposal sub-agents – a Google Place Search agent (retrieve candidate places and roads from recognised names) and an OpenStreetMap (OSM) route-matching agent (fit the route *shape* to the OSM road network) – together with a critique agent that validates each candidate against *held-out* OCR detections not used to propose it. When the critique agent confirms a route corroborated by these unused detections, that geo route, now carrying real latitude/longitude, overrides the scale-normalised trajectory. The scale-normalised route remains essential: although imperfect, its normalised scale is what makes shape-based road matching tractable. Object-anchor metric scale (for distances) and geo-grounding (for landmark identity, map-trace, and compass/directions orientation) are thus complementary.

3.2 Explore, Compile, Deploy

Explore. On the explore split – the validation set plus ∼10% of the test set – each question is handled by an autonomous agent under a high tool and token budget (quantified in Sec. 4). The explorer runs in a coding-agent harness derived from *opencode*, whose system context explicitly encourages the model to *write new scripts and tools* specialised to the question at hand – this is the origin of the synthesised map-grounding tools in Fig. 2. Within a session the agent reads the context digest; views the top-down trace, real frames, and candidate maps; runs deterministic geometry (metric line-of-sight, turn order, revisit closest-approach); calls scene-text OCR and OSM route matching; and, when a visual judgement is close, requests a second opinion from a multimodal model. It *cross-checks* each answer by reconciling at least two independent estimates (*e.g.*, object-anchor metric vs. OCR/map vs. trace vs. pixels), refining while they disagree. A session ends when the critique model's confidence exceeds ∼90% – and, on the validation set, the answer is also correct – or when the budget is exhausted, whichever comes first. Its ordered tool calls, reasoning, and final answer form a logical trace. On the unlabelled test split, where correctness is unavailable, only traces above this ∼90% confidence threshold are promoted to the compile step.

Compile. We recover the promoted traces, label each tool call by the concrete operation it performs, and take their union as a weighted transition graph. The traces make three behaviours explicit – absent from a generic plan–act loop – that we carry into the compiled policy. First, a **kind-routed operator bank** of deterministic geometry: bearing/heading and signed turn-angle (compass, directions), metric line-of-sight and object-anchor metric scale (distance, landmark), path-shape matching via resampling and dynamic time warping (DTW)/Fréchet/Procrustes (map-trace), and revisit closest-approach (revisit). Second, a set of **on-the-fly map-grounding tools** – fetch/build a reference map, overlay and render the route, align and register it to the map, and score the candidate maps – alongside the geo-grounding sub-agents (OCR, Google Place Search, OpenStreetMap matching). Third, the same **cross-check gate** the explorer already applies – commit only when at least two independent estimates agree, otherwise refine. The full set of callable tools and sub-agents is shown in Fig. 2.

Compilation turns the descriptive graph into one prescriptive policy (Fig. 1), implemented in a different harness: an Agent Development Kit (ADK) orchestrator that reuses the tools and scripts built by the explore-stage coding harness. The policy is a single agentic loop – assemble context, take an LLM step, execute a tool or sub-agent, append the result, repeat – steered by two compiled memories mined from the traces, a *kind→tool routing skill* and a set of *recurring-strategy skills* (*e.g.*, localise-then-confirm, compute–verify fusion, close-call escalation to the multimodal model, draft-then-reverify). An LLM verifier realises the cross-check gate, and a budget guard bounds the plan–act and draft-then-reverify cycles. The deterministic operators are unchanged; only routing and verification are delegated to the language model.

The compiled workflow is built *only* from these exploration traces – the synthesised tools and scripts and the logical reasoning they record: building it consumes no ground-truth labels and trains no model weights. What is amortized is problem-solving metadata – how to decompose and ground each kind of question.

Deploy. The compiled policy answers the remaining ∼90% of the test set with no per-question search, at a small fraction of the explore cost. This amortization – searching on a small split and reusing the compiled policy everywhere else – is what produces our first-place entry at deployment cost.

Every LLM call in the agent harness – in the explore-stage coding harness and the deployed ADK orchestrator alike – is served by *Gemini 3.1 Pro*; the multimodal second-opinion calls query both *Gemini 3.1 Pro* and *Gemini 3.5 Flash* [6].

4 Experiments

Setup. The test set has 986 questions over 54 videos and six kinds; the metric is top-1 leaderboard accuracy.

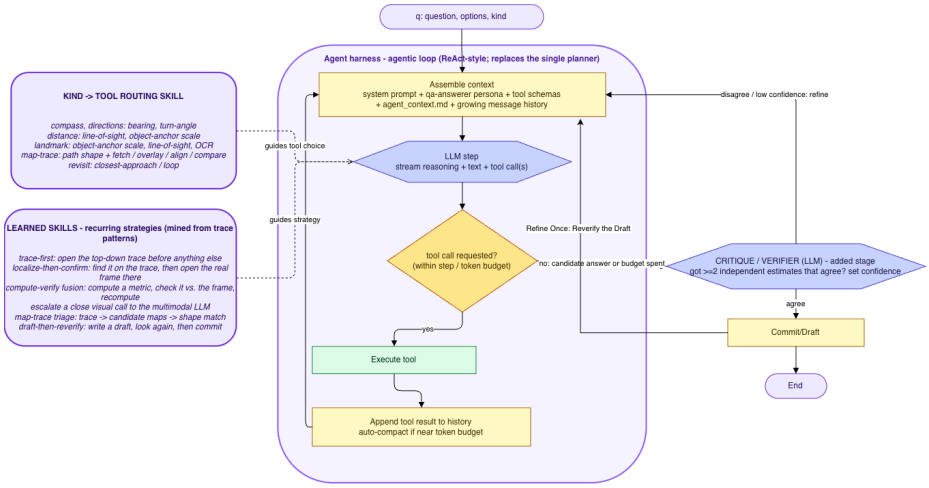

Fig. 1: Compiled policy – an ADK orchestrator induced from the explore-split traces. A single agentic loop (assemble context → LLM step → execute a tool/sub-agent → append to history) is steered by two learned memories – a kind→tool routing skill and recurring-strategy skills. An LLM verifier commits when at least two independent estimates agree, otherwise it refines; the committed draft may be re-verified once before the final answer.

Main Result. Table 1 is an additive ablation over the LLM answerer. Answering directly from the VGGT-Ω metadata, without geo-grounding, reaches 48.74%. Wrapping the LLM answerer in a critique loop (the cross-check verifier of Sec. 3.2) raises this to 68.86%, and adding geo-grounding to 78.19% – verification and map registration each contribute. Our full system, the compiled ADK orchestrator, reaches **90.4**%: first place, and 15.2 points clear of the next team on the leaderboard (75.25%).

Trace Analysis. We analyse the agent traces to quantify the explore-stage budget. The full run is mixed – it also contains lighter sessions that already reuse compiled learnings – so we characterise the explore stage by its high-effort sessions: a typical explore-stage question uses about 40 tool calls over 38 LLM steps (turns), views ∼14 frames, and takes ∼15 minutes of wall-clock time, reaching ∼70 tool calls and 33 minutes on the hardest questions; the multimodal second opinion is used in ∼25% of cases. The dominant cycle is *context* → *view imagery* → *compute* → *decide*. Cost is predictable by kind (Tab. 2): distance is cheapest, since two metric line-of-sight estimates usually agree, whereas map-trace is the heaviest – about 2× the tool calls of distance – as it requires viewing and shape-matching several candidate maps.

Labelling each tool call in these traces by the operator it performs – a view distinct from the aggregate tool/frame counts of Tab. 2 – the operators fire in strongly kind-specific patterns, which is the empirical basis for the compiled operator bank: bearing/heading is used in 100% of compass and directions ses-

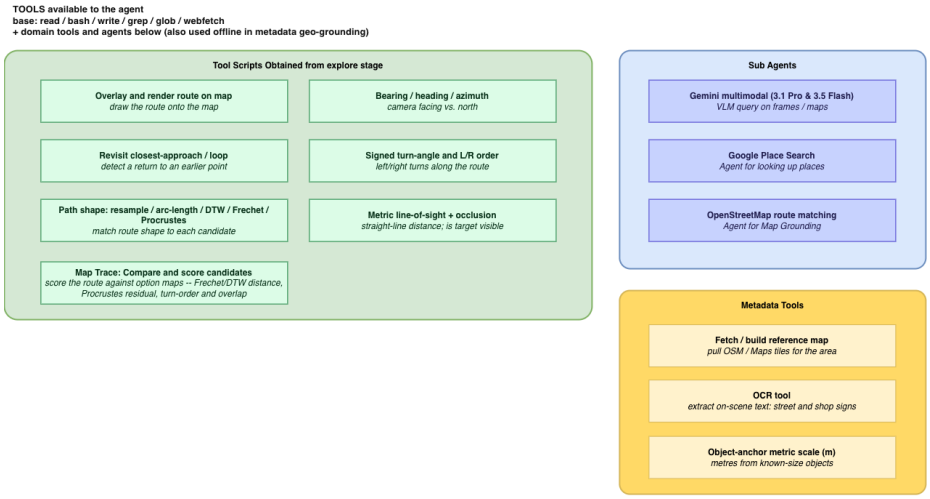

Fig. 2: Tools and sub-agents callable by the agent (blue = LLM-backed sub-agents); per-kind usage is analysed in Sec. 4. Beyond generic file/shell tools, they include frame and map viewers, a multimodal second opinion, scene-text OCR, Google Place Search and OpenStreetMap route-matching sub-agents, the deterministic geometric operators, and the synthesised map-grounding tools. Alongside the explore-stage tools, the geo-grounding sub-agents are added back into the deployed palette, and are the only tools in common with the offline metadata geo-grounding. The learned workflow skills that route these tools are shown separately in Fig. 1.

sions, object-anchor metric scale in 88% and 94% of distance and landmark sessions, synthesised map overlay/matching in 45% of map-trace sessions, and revisit closest-approach in 42% of revisit sessions. Verification is pervasive: 90% of sessions compute at least two independent estimates before committing. Answers are therefore multi-source cross-checks rather than single signals, which makes every decision auditable. On the curated metadata accuracy scales strongly with this per-question search budget, but token cost scales with it too; that cost is exactly what compilation amortizes: it is paid only on the ∼10% explore split, while the compiled policy answers the remaining ∼90% without re-searching.

5 Conclusion

We win the KilometerVision track by treating spatial grounding as amortized meta-reasoning: searching per-question grounding strategies at high effort on a small explore split with explicit verification, then compiling those traces into a single, cheap, auditable policy that generalises to the full test set. Explore-then-compile, and the trace-induced policy it yields, is a general template for grounded video reasoning whenever the answer depends on quantities a frozen model cannot read off directly.

Table 1: Test accuracy (%). Additive ablation over the LLM answerer; the first two rows use no geo-grounding. †first place; runner-up shown for reference

System	Acc.
Single LLM on metadata	48.74
+ critique loop	68.86
+ geo-grounding	78.19
Compiled ADK orch. (ours)	**90.40**†
Leaderboard runner-up	75.25

Table 2: Per-kind agent behaviour over the full run: mean tool calls and mean frames viewed

Kind	Tools	Frames
distance	14.3	4.0
compass	16.4	5.0
landmark	18.1	3.4
revisit	18.3	7.0
directions	19.7	7.3
map-trace	29.0	6.9

References

1. Cai, M., Tan, R., Zhang, J., Zou, B., Zhang, K., Yao, F., Zhu, F., Gu, J., Zhong, Y., Shang, Y., Dou, Y., Park, J., Gao, J., Lee, Y.J., Yang, J.: TemporalBench: Towards fine-grained temporal understanding for multimodal video models. arXiv preprint arXiv:2410.10818 (2024)
2. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)
3. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al.: Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. arXiv preprint arXiv:2405.21075 (2024)
4. Gershman, S.J., Goodman, N.D.: Amortized inference in probabilistic reasoning. vol. 36 (2014), https://api.semanticscholar.org/CorpusID:924780
5. Google: Gemini 3.1 Pro: A smarter model for your most complex tasks. https://blog.google/.../gemini-3-1-pro/ (2026), accessed 31 August 2026
6. Google: Gemini 3.5: Frontier intelligence with action. https://blog.google/.../gemini-3-5/ (2026), accessed 31 August 2026
7. Kesen, I., Pedrotti, A., Dogan, M., Cafagna, M., Acikgoz, E.C., Parcalabescu, L., Calixto, I., Frank, A., Gatt, A., Erdem, A., Erdem, E.: ViLMA: A Zero-Shot Benchmark for Linguistic and Temporal Grounding in Video-Language Models. In: Proceedings of the International Conference on Learning Representations (ICLR) (2024), https://openreview.net/forum?id=liuqDwmbQJ
8. Lei, J., Yu, L., Bansal, M., Berg, T.L.: TVQA: Localized, compositional video question answering. In: EMNLP (2018)
9. Li, J., Wei, P., Han, W., Fan, L.: IntentQA: Context-aware video intent reasoning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11963–11974 (2023). https://doi.org/10.1109/ICCV51070.2023.01099
10. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., Wang, L., Qiao, Y.: MVBench: A comprehensive multi-modal video understanding benchmark (2023)
11. Liang, J., Meng, X., Zhang, H., Wang, Y., Wei, J., Zhao, D.: ReasVQA: Advancing VideoQA with imperfect reasoning process. In: Proceedings of the 2025 Conference

of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). pp. 1696–1709 (2025), https://aclanthology.org/2025.naacl-long.82
12. Liu, H., Ilievski, F., Snoek, C.G.M.: Commonsense video question answering through video-grounded entailment tree reasoning. arXiv preprint arXiv:2501.05069 (2025), https://arxiv.org/abs/2501.05069
13. Liu, Y., Li, S., Liu, Y., Wang, Y., Ren, S., Li, L., Chen, S., Sun, X., Hou, L.: TempCompass: Do video LLMs really understand videos? arXiv preprint arXiv:2403.00476 (2024)
14. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. vol. 36, pp. 46212–46244 (2023)
15. Meng, J., Li, X., Wang, H., Tan, Y., Zhang, T., Kong, L., Tong, Y., Wang, A., Teng, Z., Wang, Y., Wang, Z.: Open-o3 video: Grounded video reasoning with explicit spatio-temporal evidence. arXiv preprint arXiv:2510.20579 (2025), https://arxiv.org/abs/2510.20579
16. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
17. Rawal, R., Saifullah, K., Basri, R., Jacobs, D., Somepalli, G., Goldstein, T.: CinePile: A long video question answering dataset and benchmark. arXiv preprint arXiv:2405.08813 (2024)
18. Russell, S., Wefald, E.: Principles of metareasoning. Artificial Intelligence **49**(1), 361–395 (1991). https://doi.org/https://doi.org/10.1016/0004-3702(91)90015-C, https://www.sciencedirect.com/science/article/pii/000437029190015C
19. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language models can teach themselves to use tools. vol. 36, pp. 68539–68551 (2023)
20. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
21. Shangguan, Z., Li, C., Ding, Y., Zheng, Y., Zhao, Y., Fitzgerald, T., Cohan, A.: Tomato: Assessing visual temporal reasoning capabilities in multimodal foundation models. arXiv preprint arXiv:2410.23266 (2024)
22. Swetha, S., Gupta, R., Kulkarni, P.P., Shatwell, D.G., A Chan Santiago, J., Siddiqui, N., Fioresi, J., Shah, M.: VRR-QA: Visual relational reasoning in videos beyond explicit cues. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 32840–32849 (June 2026)
23. Swetha, S., Kuehne, H., Shah, M.: TimeLogic: A temporal logic benchmark for video QA. arXiv preprint arXiv:2501.07214 (2025)
24. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5294–5306. IEEE (2025)
25. Wang, J., Chen, M., Zhang, S., Karaev, N., Schönberger, J., Labatut, P., Bojanowski, P., Novotny, D., Vedaldi, A., Rupprecht, C.: VGGT-Ω. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
26. Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R.K.W., Lim, E.P.: Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language

models. In: Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers). pp. 2609–2634 (2023)
27. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. vol. 35, pp. 24824–24837 (2022)
28. Wu, B., Yu, S., Chen, Z., Tenenbaum, J.B., Gan, C.: STAR: A benchmark for situated reasoning in real-world videos. In: Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS) (2021)
29. Xiao, J., Shang, X., Yao, A., Chua, T.S.: NExT-QA: Next phase of question-answering to explaining temporal actions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9777–9786 (June 2021)
30. Xu, J., Mei, T., Yao, T., Rui, Y.: MSR-VTT: A large video description dataset for bridging video and language. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
31. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: React: Synergizing reasoning and acting in language models (2022)
32. Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., Tao, D.: Activitynet-qa: A dataset for understanding complex web videos via question answering. In: AAAI. pp. 9127–9134 (2019)