

EgoTL: Egocentric Think-Aloud Chains for Long-Horizon Tasks

Supplementary Material

This supplement is organized as follows:

- Section A details the benchmark construction pipeline, including dataset curation, Q&A construction, clip-wise labeling, and 3D trajectory-based distance annotation. Section A.1 lists the details about the Q&A construction. Section A.2 lists the full prompt templates used for all six evaluation tasks, covering memory-grounded planning, action reasoning, next-action prediction, action recognition, direction recognition, and distance estimation.
- Section B describes the evaluation setup for closed-source VLMs and how we adapt our benchmark to their interfaces and input constraints.
- Section C provides VLM fine-tuning specifications, including model architectures and checkpoints (Section C.1), training data configuration (Section C.2) and evaluation (Section C.3).

A. Benchmark Construction Details

A.1. Q&A Construction

For this step, we use Gemini 2.5 Flash [2] to automatically synthesize three distractor options for each multiple-choice question. We first sample 100 representative Q&A pairs from our tasks and use them as a pilot set. For these pairs, we prompt Gemini 2.5 Flash [2] to generate three distractor options per question. Human annotators then carefully review the generated options to evaluate their quality: they check that each distractor is (1) clearly incorrect yet still plausible given the question context, (2) linguistically clear and unambiguous, and (3) free of hallucinated content or information that cannot be inferred from the provided input. Based on this manual review, we iteratively refine the prompting strategy (e.g., by specifying stricter constraints on correctness, relevance, and style of the options) until annotators are satisfied that the model reliably produces high-quality distractors on the pilot set. Once this prompt design is stabilized, we apply the same prompting pipeline to all Q&A pairs in our tasks to generate three distractor options for each question at scale. Each prompt is tailored to its corresponding task in order to match the task-specific context and reasoning requirements.

A.2. Prompt Templates for the Six Tasks

For completeness, we list the generic prompt schemas used in all experiments.

A.2.1. Memory-Grounded Task Planning

For the memory-bank evaluation, we prompt the VLMs with the abstract task description, the corresponding

memory-bank video, and four candidate chains-of-thought (options A–D). The model is instructed that the options are shuffled and that it must read and compare all four options. We then ask the model to assign each option a real-valued score between 0 and 1, indicating how well that option matches the given task and memory-bank video, and to output only a JSON object of the form “scores”: “A”: 0.0, “B”: 0.0, “C”: 0.0, “D”: 0.0, “best”: “A”. Here, “scores” is used to encourage the model to perform fine-grained, relative comparison across all options, while “best” denotes the single selected option (A/B/C/D). During evaluation, we parse the “best” field from the JSON output and compare the corresponding option against the ground-truth CoT label to compute accuracy; the per-option scores are treated as auxiliary signals and are not used directly in the metric.

A.2.2. Action Reasoning under Complex Environment

For the action reasoning task, we probe whether VLMs can choose scene-aware actions in cluttered, physically constrained environments. Each instance consists of an abstract task description, a short egocentric video clip showing the current state of the environment (including obstacles and free space), and four candidate reasoning steps (options A–D) that describe what the camera wearer should do next. The candidates are constructed so that only one option is globally consistent with both the task and the physical layout (e.g., avoiding blocked paths or unreachable objects), while the others either ignore obstacles, violate basic physical constraints, or are task-irrelevant.

A.2.3. Next-Action Prediction

Prompt for next-action prediction. For the next-action evaluation, we condition the VLMs on the current egocentric video clip and four candidate next-action descriptions. The model is instructed that it is an egocentric next-action predictor, that it will see the current clip, and that it must choose exactly one next action from the candidates. The prompt explicitly asks the model to output only the chosen option text (verbatim), without any additional explanation or formatting. During evaluation, we parse the model’s response by matching the returned text against the four candidates and treat the matched candidate as the predicted next action. Accuracy is computed by comparing this predicted option with the ground-truth next-action label derived from the original (unshuffled) CSV, while any extra text beyond the selected option is ignored.

A.2.4. Action Recognition

For the action-recognition benchmark, we evaluate the VLMs using short egocentric video clips paired with four

Task	Question Template	Input
Planning from Memory	<i>You are given an egocentric task {abstract_task} and four candidate chains-of-thought (A, B, C, D). Options are shuffled; you MUST read and compare all options and assign each a score between 0 and 1 according to how well it matches the task and the memory-bank video {memory_bank}. Then select the single best option and return ONLY a JSON object of the form { ``scores``: { ``A``: 0.0, ``B``: 0.0, ``C``: 0.0, ``D``: 0.0}, ``best``: ``A``}.</i>	<i>Memory-bank video + text prompt</i>
Action Reasoning under Complex Environment	<i>The image shows the current condition and you are given an egocentric {abstract_task}. You are an egocentric action reasoning agent: based on the scene, decide what the camera wearer should do next while considering obstacles, free space, and object locations. You are given four candidate next actions (A–D). Carefully read the task and analyze the image, then select the single option that best accomplishes the task and is physically feasible in the scene. Output only the text of the chosen option, exactly as written, with no additional explanation or formatting.</i>	<i>Current frame + abstract task</i>
Next Action Recognition	<i>You are an egocentric next-action predictor. You will see the current egocentric video clip and a chain-of-thought describing the ongoing task {CoT}. From the candidate next actions ({option 1}, {option 2}, {option 3}, {option 4}), choose exactly one next action and output only the chosen option text.</i>	<i>Current video clip + CoT text</i>
Action Recognition	<i>You are an egocentric video action classifier. You will be given a short egocentric video clip and four candidate descriptions of the action, each labeled with a letter: A, B, C, and D. Select the one option that best matches the action shown in the video and answer with exactly one capital letter (A, B, C, or D) and nothing else. The four options are: {option 1}, {option 2}, {option 3}, {option 4}.</i>	<i>Current video clip</i>
Direction Recognition	<i>You are a video grounding agent. From the egocentric video, choose the dominant motion direction: Turn left, Turn right, Move forward, Going up, or Going down. Decide based on the global motion trend and ignore small local jitters. Output only the text of the chosen option (not the letter).</i>	<i>Current video clip</i>
Distance Estimation	<i>You are a video measurement agent. From the egocentric video, estimate the approximate distance traveled in meters by the dominant actor or camera. Use global motion and scene-scale cues, and ignore small jitter or in-place head movements. If there is essentially no movement, output 0. Answer with a single real-valued number in meters and nothing else.</i>	<i>Current video clip</i>

Table 1. **Question Templates for tasks in EgoTL-Bench.** We replace the **highlighted** part in the question template from scene to scene to construct our benchmark.

candidate action descriptions. Each candidate is labeled with a letter (A, B, C, or D), and the model is prompted that it is an egocentric video action classifier. The prompt presents the current video clip together with the four labeled options and instructs the model to select exactly one option that best matches the action shown in the video and to answer with only a single capital letter (A, B, C, or D), without any additional text.

Given the model’s text output, we parse the first valid capital letter in {A, B, C, D} and treat it as the predicted label. The ground-truth action for each clip is obtained from a separate CSV file and mapped to one of the four options by string normalization and exact matching; this defines the gold letter (A-D). We then compare the model’s predicted letter with the gold letter to compute accuracy. If the model output does not contain any valid letter, the sample

is recorded but excluded from the scored set.

A.2.5. Direction Recognition

For direction recognition, we evaluate the VLMs on short egocentric video clips labeled with one of five motion directions: *Turn left*, *Turn right*, *Move forward*, *Going up*, or *Going down*. The model is prompted as a video grounding agent and instructed to pick the dominant motion direction from this set: it must rely on the global camera/actor motion and explicitly ignore small jitter or head movements. The prompt lists the five options in natural language and asks the model to output only the *text* of the chosen option (e.g., “Turn left”).

At evaluation time, we normalize the model’s free-form text response by lowercasing, stripping punctuation, and mapping common paraphrases (e.g., “go forward”, “move forward”) to a standard label using a keyword table. We then compare this prediction with the ground-truth direction for each clip to compute overall accuracy as well as per-direction accuracy.

A.2.6. Distance Estimation

For distance estimation, we ask the VLMs to infer how far the camera (or dominant actor) has moved in each egocentric clip, measured in meters. The model is prompted as a video measurement agent and instructed to estimate the approximate traveled distance using global motion and scene-scale cues while ignoring small jitter or in-place head movements; if the scene is essentially stationary, it should output 0. The prompt explicitly requires the model to respond with a single number in meters and no additional text.

For each video, we obtain a scalar ground-truth distance from pre-computed annotations stored in text files and parse the model’s response by extracting the first floating-point number as the predicted distance. We then compute the absolute error, the relative error, and a mean relative accuracy (MRA) score [7]: for a set of thresholds $\{\theta\}$ in $[0.5, 0.95]$, we check whether the relative error is below $1 - \theta$ and average the resulting binary indicators across thresholds. This MRA metric rewards predictions that stay consistently close to the ground-truth distance under progressively stricter tolerance levels.

B. Closed-Source Benchmark Setup

We evaluate several closed-source VLMs, including GPT-5 [1], GPT-4o [4], Gemini 2.0 Flash [5], and Gemini 2.5 Flash [2], on our EgoTL-Bench and compare them with open-source baselines. On high-level tasks such as memory-conditioned planning and scene-aware action reasoning, all models remain far below human performance, indicating that long-horizon planning is still challenging. Within this low absolute regime, closed-source systems consistently achieve higher scores than open-source models, suggesting

a relative advantage in high-level reasoning. At the perceptual layer, however, this gap narrows or even reverses: on next-action prediction and action recognition, strong open-source models are often comparable to, or slightly stronger than, closed-source ones. For direction recognition, both open-source and closed-source models exhibit a strong bias toward predicting “Move forward”, which makes it difficult for them to reliably distinguish turning motions. Moreover, almost all models perform poorly on distance estimation, indicating that current VLMs still lack robust egocentric distance understanding. Overall, these results show that current VLMs remain far from human-level egocentric spatial understanding and long-horizon reasoning, and that substantial progress is still required.

C. VLM Fine-Tuning Specification

C.1. Model Architectures and Checkpoints

We fine-tune Qwen2.5-VL-7B-Instruct [6], a 7B-parameter multimodal vision–language transformer with a frozen vision encoder and a language backbone augmented with cross-modal adapters. To adapt the model to EgoTL without overfitting or incurring the full cost of dense fine-tuning, we adopt low-rank adaptation (LoRA) [3] on top of the language backbone. Specifically, we insert rank-16 LoRA adapters into all transformer blocks while keeping both the vision tower and the multimodal projector frozen. This design allows the model to specialize to EgoTL’s spatial reasoning distribution while preserving the strong general-purpose capabilities of the base checkpoint.

C.2. Training Data Configuration

We use a disjoint subset of EgoTL to fine-tune our VLM and then use the test set to evaluate the model. The test set contains 100 task videos spanning 15 scenes. After applying the same curation pipeline as in the main benchmark, we obtain 1.2k Q&A pairs for training.

C.3. Evaluation

We evaluate the fine-tuned model on the same test set described above. Our model surpasses the strongest baseline across all layers and metrics, with particularly notable gains in high-level planning. At the low-level perceptual layer, it also consistently achieves better performance. In particular, for distance estimation, where all current VLMs struggle, our fine-tuned model attains substantially higher mean relative accuracy, nearly doubling the MRA of the best pre-fine-tuning configuration. These improvements suggest that our human-annotated dataset not only provides VLMs with implicit scale calibration, but also offers reliable supervision, demonstrating that carefully collected human data can substantially improve VLM spatial reasoning.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3
- [2] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blisstein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 1, 3
- [3] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3
- [4] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv*, 2024. 3
- [5] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 3
- [6] Qwen Team. Qwen2.5: A party of foundation models, 2024. 3
- [7] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. *arXiv preprint arXiv:2412.14171*, 2024. 3