MET-Bench: Multimodal Entity Tracking for Evaluating the Limitations of Vision-Language and Reasoning Models

Vanya Cohen [1] **Raymond Mooney** [1]

Abstract

We introduce MET-Bench, a multimodal entity tracking benchmark designed to evaluate the ability of vision-language models to track entity states across modalities. Using two structured domains, Chess and the Shell Game, we assess how frontier models integrate textual and image-based state updates. Our findings reveal a significant performance gap between text-based and image-based tracking. We show this performance gap stems from deficits in visual reasoning rather than perception and that explicit text-based reasoning strategies improve performance, yet limitations remain, especially in long-horizon multimodal scenarios. MET-Bench highlights the need for improved multimodal representations and reasoning techniques to bridge the gap between textual and visual entity tracking.

1. Introduction

World understanding requires tracking information about entity state as it evolves through text, images, videos and other modalities. Our work examines this challenge through the lens of multimodal entity state tracking, where changes to entity states must be understood from both textual descriptions and visual observations. This setting provides a natural extension to classical NLP problems related to textual entity tracking while connecting to emerging research in world models.

We introduce MET-Bench to assess how effectively current language models can track entity states when updates are conveyed through both text and images. We find that current language models struggle with multimodal entity tracking not due to low-level perceptual failures but because they lack representations for updating entity state across sequential visual observations. This suggests a fundamental limitation in how these models integrate and update state representations from different modalities.

We make the following contributions:

- We introduce the multimodal entity tracking benchmark (MET-Bench) that extends traditional NLP entity tracking evaluation to the multimodal setting for two domains: *multimodal Chess* and *Shell Game*.
- We demonstrate that current models, despite strong performance on pure text tasks, struggle to maintain accurate entity representations when processing mixed text and image inputs.
- Through experiments, we show that these limitations stem from higher-level reasoning challenges rather than low-level perception issues.
- We evaluate various approaches to improving multimodal entity tracking, finding that techniques emphasizing explicit reasoning outperform methods that rely on more training examples, especially when generalizing to novel domains.

2. Methods

We formulate the problem of *multimodal entity tracking* as a sequential state estimation task, where an agent must infer the final state of a system given an initial state and a series of observed actions. MET-Bench represents the initial and final states of each domain as text but evaluate the models' ability to track entity state changes through images. This approach isolates the multimodal entity tracking challenge by ensuring that models begin and end with well-defined textual representations to confounding errors from perceptual failures, which remain a known limitation of current vision-language models (Sharma et al., 2024). We construct two domains for evaluating model performance.

Chess Domain Chess is a well-studied domain for testing entity tracking of deep learning models (Toshniwal et al., 2022). The entity state is an 8×8 board configuration expressed in Forsyth–Edwards Notation (FEN) notation, actions correspond to legal chess moves from real games, and action observations consist of either symbolic (UCI

*Equal contribution [1]Department of Computer Science, The University of Texas at Austin. Correspondence to: Vanya Cohen <vanya@utexas.edu>.

ICML 2025 Workshop on Assessing World Models.

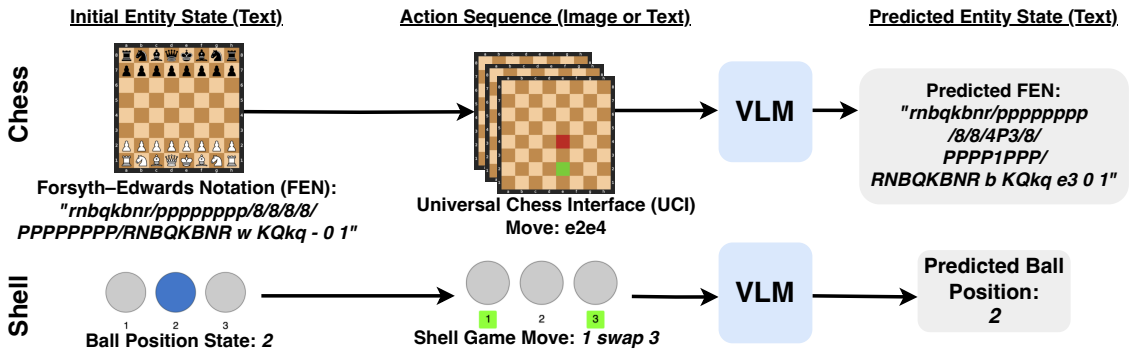


Figure 1: In the multimodal entity tracking benchmark (MET-Bench), a vision-language model (VLM) predicts the final entity state from the initial state and actions which update the entity state. The initial entity state is provided as text and the actions are given as images or as text. The predicted final state is a text representation of the entity state. The image and text representations in this figure are are used for evaluation.

Role	Messages

User	`You are a helpful assistant that tracks chess moves in a game and produces the final FEN. The initial state is:`
	`rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1`
	`Here are the moves played:`
	`e2e4`
	`e7e5`
	`Now what is the final FEN? Output FINAL ANSWER: [FEN].`

Assistant	`FINAL ANSWER: rnbqkbnr/pppp1ppp/8/4p3/4P3/8/PPPP1PPP/RNBQKBNR w KQkq - 2 2`

Figure 2: An example zero-shot user–assistant exchange in the **Chess** domain, showing the initial board state as FEN, two UCI moves (e2e4, e7e5) and the final state. For image actions, the UCI moves are replaced with their visual representations and a description of how to interpret these images. The FEN is line-broken for readability. For details see Appendix A, Figures 7 & 8.

notation) or visual (board images) descriptions of moves. Utilizing real Chess games from the Millionbase dataset[1] used in Toshniwal et al. (2022), we generate sequences of states and actions (moves) using standard chess notation: Universal Chess Interface (UCI) for actions and FEN for board states.

Shell Game Domain Shell Game is classic demonstration of hidden-state tracking. A ball is placed under one of three cups (or shells), which are then swapped pairwise in succession. The goal is tracking which cup currently hides the ball as shells are swapped. The state is the hidden position of a ball under three shells and actions correspond to swaps between pairs of shells. Other works have explored shell-game-like domains with varying levels of added complexity (Li et al., 2021; Long et al., 2016; Kim & Schuster, 2023). These image representations were created through visual-prompt engineering to maximize the classification accuracy of actions depicted.

[1] https://rebel13.nl/rebel13/rebel%2013.html

3. Experiments

Tracking in Text Outperforms Images We evaluate difference in accuracy when tracking images from text and image actions in the zero-shot, few-shot, and chain-of-thought, and reasoning settings in Table 1. We evaluate on a set of 500 games selected at random from the test set, each with a sequence length of ten actions. Across both domains and all models, entity tracking in text outperforms tracking in images, with the exception of Gemini 2.5 Pro which attains equal performance in these modalities.

Reasoning Aids Long Sequence Accuracy We evaluate longer sequences ranging to 100 actions in both the text modality and 20 in the image modality. Results for Chess are in Figure 3 and Shell Game in Figure 4. We evaluate reasoning models and other frontier models in the chain-of-thought setting. Reasoning models tend to perform better on these longer sequences. Most frontier models perform at baseline level in the image modality.

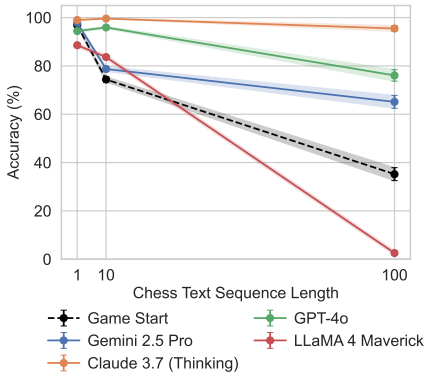
Models Understand Image Actions We perform an experiment to demonstrate that VLMs have the ability to accurately interpret the actions depicted in the image-action representations. Table 2 shows the performance on clas-

Table 1: Entity tracking accuracy (95% CI) in Chess and Shell Game for text-only and image-only actions. In the Few-Shot setting $N = 5$ in-context examples are used. Methods with explicit reasoning perform best. The baseline of predicting the Game Start board is given for Chess, and randomly selecting a final state for Shell Game.

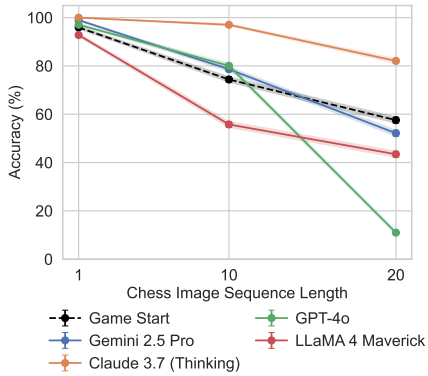
Model	Chess		Shell	
	Text	**Image**	**Text**	**Image**
Baseline				
Game Start, Random	74.9 ± 0.5	74.9 ± 0.5	33.0	33.0
Zero-Shot				
GPT-4o	91.6 ± 0.3	76.0 ± 0.5	33.0 ± 4.1	32.2 ± 4.1
GPT-4o-mini	75.6 ± 0.5	55.3 ± 0.5	33.6 ± 4.1	32.4 ± 4.1
GPT-4.1	85.6 ± 0.4	67.7 ± 0.5	32.2 ± 4.1	36.4 ± 4.2
GPT-4.1-mini	82.5 ± 0.4	77.5 ± 0.5	31.4 ± 4.1	30.8 ± 4.0
GPT-4.1-nano	77.5 ± 0.5	73.3 ± 0.5	31.2 ± 4.0	30.8 ± 4.0
Gemini-2.5-Flash	91.0 ± 0.3	66.9 ± 0.5	35.0 ± 4.2	37.0 ± 4.2
LLaMA-4 Maverick	86.7 ± 0.4	51.1 ± 1.2	30.8 ± 4.0	33.2 ± 4.1
MiniMax-VL-01	85.9 ± 0.4	73.4 ± 0.5	30.8 ± 4.0	31.2 ± 4.0
Claude 3.7 Sonnet	96.1 ± 0.2	70.2 ± 0.5	35.4 ± 4.2	37.8 ± 4.2
Few-Shot (n=5)				
GPT-4o	94.3 ± 0.2	77.6 ± 0.5	33.6 ± 4.1	32.0 ± 4.1
GPT-4o-mini	74.5 ± 0.5	74.2 ± 0.5	34.8 ± 4.2	32.2 ± 4.1
GPT-4.1	86.6 ± 0.4	64.9 ± 0.5	34.6 ± 4.2	33.0 ± 4.1
GPT-4.1-mini	81.2 ± 0.4	76.9 ± 0.5	36.4 ± 4.2	32.8 ± 4.1
GPT-4.1-nano	74.5 ± 0.5	74.6 ± 0.5	36.6 ± 4.2	36.4 ± 4.2
Gemini-2.5-Flash	91.3 ± 0.3	72.0 ± 0.5	31.4 ± 4.1	35.2 ± 4.2
LLaMA-4 Maverick	85.8 ± 0.4	48.1 ± 0.6	35.4 ± 4.2	35.2 ± 4.2
MiniMax-VL-01	88.0 ± 0.8	40.8 ± 1.2	36.4 ± 4.2	32.0 ± 4.1
Claude 3.7 Sonnet	99.2 ± 0.1	77.7 ± 0.5	32.4 ± 4.1	36.0 ± 4.2
Chain-of-Thought				
GPT-4o	94.9 ± 0.2	67.5 ± 0.5	**99.0 ± 0.9**	35.8 ± 4.2
GPT-4o-mini	68.8 ± 0.5	43.1 ± 0.5	61.0 ± 4.3	30.4 ± 4.0
GPT-4.1	98.1 ± 0.1	75.3 ± 0.5	**99.8 ± 0.5**	37.6 ± 4.2
GPT-4.1-mini	86.4 ± 0.4	77.6 ± 0.5	**100.0 - 0.4**	72.0 ± 3.9
GPT-4.1-nano	49.0 ± 0.6	34.6 ± 0.5	61.8 ± 4.2	32.2 ± 4.1
Gemini-2.5-Flash	77.0 ± 1.0	44.6 ± 1.2	94.0 ± 4.8	34.0 ± 9.1
LLaMA-4 Maverick	82.4 ± 0.4	61.9 ± 0.5	77.0 ± 3.7	34.6 ± 4.2
MiniMax-VL-01	62.3 ± 0.5	32.8 ± 0.5	77.4 ± 3.7	34.4 ± 4.2
Claude 3.7 Sonnet	99.5 ± 0.1	**96.2 ± 0.2**	**100.0 - 0.4**	77.4 ± 3.7
Reasoning				
o1	98.2 ± 0.3	83.5 ± 0.9	**100.0 - 1.9**	92.6 ± 2.3
o3	**99.9 ± 0.1**	45.5 ± 1.2	**100.0 - 1.9**	63.0 ± 9.3
o4-mini	84.2 ± 0.9	78.0 ± 1.0	**100.0 - 1.9**	33.0 ± 9.1
Gemini-2.5-Pro	77.4 ± 1.0	76.8 ± 1.0	**100.0 - 1.9**	**100.0 - 1.9**
Gemini-2.5-Flash (thinking)	40.2 ± 1.2	17.4 ± 0.9	**100.0 - 1.9**	42.0 ± 9.5
Claude 3.7 Sonnet (thinking)	**99.8 ± 0.1**	**96.0 ± 0.5**	**100.0 - 1.9**	87.0 ± 6.6

sifying the text action represented by each image action. We evaluate the recognition of the start (e.g. piece moved), end, and 'Overall' accuracy of classifying the entire action (start and end) correctly. GPT-4o achieves an accuracy of $95.2\% \pm 0.4\%$ in Chess and all models attain perfect accuracy on the simpler Shell Game domain. This indicates that perception of the image-actions is not the fundamental limiting factor for effective entity tracking with image inputs.

Cascading Matches Text-Only Tracking Using the text actions predicted from the images in the image-action classification task, we devise an ablation to test the effect of cascading (first captioning the image actions, and then tracking entities purely in text). Table 3 shows the accuracy of cascaded inference. The performance in the cascaded setting is similar to the text-action performance, showing that the model has the task-knowledge needed to perform entity tracking in both domains, but cannot reason effectively in the image modality.

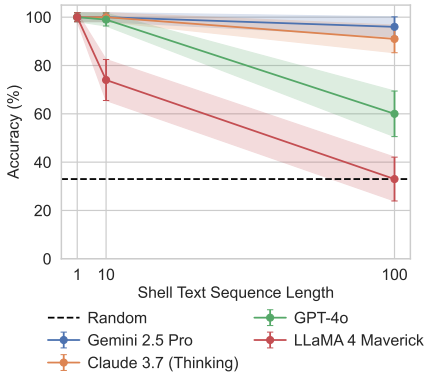


(a) Chess accuracy with text actions.

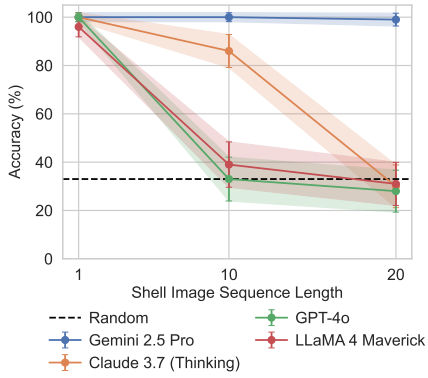


(b) Chess accuracy with image actions.

Figure 3: In the text action setting, the reasoning models Claude 3.7 Sonnet Thinking and Gemini 2.5 Pro, maintain the highest accuracy at longer sequence lengths. All models struggle to maintain accurate board representations in the image action setting, with Claude 3.7 Sonnet Thinking performing the best. 95% confidence intervals.



(a) Shell Game accuracy with text actions.



(b) Shell Game accuracy with image actions.

Figure 4: In the text action setting, the reasoning models Claude 3.7 Sonnet Thinking and Gemini 2.5 Pro achieve the highest performance over long action sequences. In the image action setting the accuracy of all models except Gemini 2.5 Pro decreases to random by 20 actions. 95% confidence intervals.

Table 2: Percent image-action classification accuracy (95% CI) for various models. We report the accuracy of predicting the action start, end, and overall/UCI action for both Chess and Shell Game on 10,000 image actions.

Model	Start (%)	End (%)	Overall (%)
Chess			
GPT-4o-mini	67.2 ± 0.9	57.4 ± 1.0	46.4 ± 1.0
GPT-4o	96.4 ± 0.4	98.6 ± 0.2	95.2 ± 0.4
Shell Game			
GPT-4o-mini	100.0 - 0.2	100.0 - 0.2	100.0 - 0.2
GPT-4o	100.0 - 0.2	100.0 - 0.2	100.0 - 0.2

Table 3: Cascaded entity tracking accuracy (95% CI) for Chess and Shell (Image → Text). In cascaded inference, the model is first used to map each image action to the text representation of the action. Then model is prompted to perform the entity tracking task as in the text-action setting.

	Cascaded	
Method	Chess	Shell
Chain-of-Thought		
GPT-4o mini	66.4 ± 0.4	47.6 ± 3.1
GPT-4o	93.2 ± 0.2	99.4 ± 0.5

4. Discussion

Our evaluation of frontier model performance on MET-Bench provides several insights into the current state and remaining challenges of multimodal entity tracking. We demonstrate a significant performance gap between text-based and image-based entity tracking across all evaluated models, with even state-of-the-art vision-language-reasoning models struggling to maintain accurate entity

states when processing visual inputs. This disparity persists across both the Chess and Shell Game domains, suggesting a fundamental limitation in current architectures' ability to reason about entity states through visual observations.

This finding is particularly noteworthy given that our image-action classification results (Table 2) demonstrate that models can accurately perceive and classify individual visual actions. The gap between perception and reasoning suggests that the challenge lies not in processing visual inputs, but in maintaining and updating coherent entity information across sequential visual observations.

Our cascaded inference experiments provide further evidence for this interpretation. When models first translate visual inputs to text before performing entity tracking, they achieve performance comparable to pure text-based tracking. This indicates that the models possess the relevant task knowledge and reasoning capabilities, but struggle to apply them directly in the visual domain.

Further, the effectiveness of chain-of-thought prompting, particularly in the Shell Game domain where it improved Claude 3.7 Sonnet's accuracy from near random to $100.0\% - 0.4\%$ for text and $77.4\% \pm 3.7\%$ for images, highlights the importance of explicit reasoning for entity tracking. This improvement indicates that current models can perform complex entity tracking when guided to decompose the task into smaller steps, even in novel domains not present in their training data. However, the fact that such prompting was necessary suggests that models do not implement robust tracking, particularly in multimodal settings. Lastly, the performance of specialized reasoning models like Gemini 2.5 Pro and Claude 3.7 Sonnet Thinking on longer sequences demonstrates the potential of architectures explicitly trained for sequential reasoning to maintain coherent entity states despite the challenges of accumulating errors over extended sequences.

5. Related Work

Entity tracking has been extensively studied in textual domains, with a focus on probing and improving language models' abilities to maintain representations of entity states. For instance, Toshniwal et al. (2022) evaluates chess as an entity tracking domain, employing fine-tuned models (Radford et al., 2019) to assess performance. Similarly, Kim & Schuster (2023) examine the impact of model size and fine-tuning on entity tracking in textual settings similar to our Shell Game domain. Tandon et al. (2020) construct a benchmark for understanding entity state changes in procedural texts. Shirai et al. (2022) construct the Visual Recipe Flow corpus and evaluate the ability of multimodal embedding models to properly sequence images depicting recipe states. In contrast, our work requires predicting entity state changes from actions specified in images and involves larger state spaces.

Several studies explore the implicit representations of entity states in language models. Li et al. (2021) and Long et al. (2016) use semantic probing to reveal that Transformer-based models (Vaswani et al., 2017) capture entity state representations implicitly during textual reasoning. Building on this, Prakash et al. (2024) demonstrate that fine-tuning language models for entity tracking tasks enhances pre-existing internal mechanisms rather than learning entirely new representations. Li et al. (2023) find that Transformers trained on Othello games form internal representations of the game state.

Efforts to improve textual entity tracking beyond domain-specific fine-tuning include Fagnou et al. (2024), which establishes theoretical limitations of the Transformer architecture in tracking entities. They propose a novel attention mechanism to enhance entity tracking in Transformers. Gupta & Durrett (2019) fine-tunes small Transformer-based models for tracking entity state in instructional texts. Kim et al. (2024) investigates how code pretraining improves language models' abilities to track entities in text, while Yoneda et al. (2024) introduce Stalter, a prompting method designed to maintain accurate state representations in text-based robotics planning.

These works focus on entity tracking as a unimodal, text-based reasoning task. While unimodal approaches have achieved substantial progress, there remains a gap in evaluating models' ability to integrate multimodal inputs for entity tracking. Our work extends these evaluations to the multimodal setting and quantifies the performance improvement of reasoning models for entity tracking.

6. Conclusion

Our findings suggest that the primary bottleneck in multimodal entity tracking is not visual recognition but sequential reasoning over visual updates. Unlike text-based representations, which align with the models' training paradigms, visual updates require implicit state reconstruction—a task that current architectures do not perform reliably. Future work should explore the effect of additional visual-reasoning post-training, explicit memory structures, or hybrid symbolic representations to mitigate this gap. Additional research directions include investigating the role of entity tracking in world-modeling, narrative understanding, and expanding MET-Bench to include more complex domains beyond games. We believe addressing these challenges will be crucial for developing AI systems capable of robust reasoning for real-world tasks.

References

Anthropic. Claude 3.7 Sonnet. Anthropic Announcement (Feb. 24, 2025), 2025. URL https://www.anthropic.com/news/claude-3-7-sonnet.

DeepMind. Gemini 2.5 Pro. Google DeepMind (March. 25, 2025), 2024. URL https://storage.googleapis.com/model-cards/documents/gemini-2.5-pro-preview.pdf.

Fagnou, E., Caillon, P., Delattre, B., and Allauzen, A. Chain and causal attention for efficient entity tracking. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13174–13188, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.731. URL https://aclanthology.org/2024.emnlp-main.731/.

Gupta, A. and Durrett, G. Effective use of transformer networks for entity tracking. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 759–769, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1070. URL https://aclanthology.org/D19-1070/.

Kim, N. and Schuster, S. Entity tracking in language models. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3835–3855, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.213. URL https://aclanthology.org/2023.acl-long.213/.

Kim, N., Schuster, S., and Toshniwal, S. Code pretraining improves entity tracking abilities of language models. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2405.21068.

Li, B. Z., Nye, M., and Andreas, J. Implicit representations of meaning in neural language models. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1813–1827, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143. URL https://aclanthology.org/2021.acl-long.143/.

Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., and Wattenberg, M. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DeG07_TcZvT.

Long, R., Pasupat, P., and Liang, P. Simpler context-dependent logical forms via model projections. In Erk, K. and Smith, N. A. (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1456–1465, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1138. URL https://aclanthology.org/P16-1138/.

Meta. Llama 4 Model Card. OpenAI Technical Report (April 5, 2025), 2025. URL https://github.com/meta-llama/llama-models/blob/main/models/llama4/MODEL_CARD.md.

MiniMax, Li, A., Gong, B., Yang, B., Shan, B., Liu, C., Zhu, C., Zhang, C., Guo, C., Chen, D., Li, D., Jiao, E., Li, G., Zhang, G., Sun, H., Dong, H., Zhu, J., Zhuang, J., Song, J., Zhu, J., Han, J., Li, J., Xie, J., Xu, J., Yan, J., Zhang, K., Xiao, K., Kang, K., Han, L., Wang, L., Yu, L., Feng, L., Zheng, L., Chai, L., Xing, L., Ju, M., Chi, M., Zhang, M., Huang, P., Niu, P., Li, P., Zhao, P., Yang, Q., Xu, Q., Wang, Q., Wang, Q., Li, Q., Leng, R., Shi, S., Yu, S., Li, S., Zhu, S., Huang, T., Liang, T., Sun, W., Sun, W., Cheng, W., Li, W., Song, X., Su, X., Han, X., Zhang, X., Hou, X., Min, X., Zou, X., Shen, X., Gong, Y., Zhu, Y., Zhou, Y., Zhong, Y., Hu, Y., Fan, Y., Yu, Y., Yang, Y., Li, Y., Huang, Y., Li, Y., Huang, Y., Xu, Y., Mao, Y., Li, Z., Li, Z., Tao, Z., Ying, Z., Cong, Z., Qin, Z., Fan, Z., Yu, Z., Jiang, Z., and Wu, Z. Minimax-01: Scaling foundation models with lightning attention, 2025. URL https://arxiv.org/abs/2501.08313.

OpenAI. GPT-4o mini: advancing cost-efficient intelligence. OpenAI Blog (July 18, 2024), 2024a. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence.

OpenAI. Hello GPT-4o. OpenAI Announcement (May 13, 2024), 2024b. URL https://openai.com/index/hello-gpt-4o.

OpenAI. Introducing OpenAI o1-preview and o1-mini. OpenAI Release Notes (Sept. 12, 2024), 2024c. URL https://help.openai.com/en/articles/9624314-model-release-notes.

OpenAI. Introducing GPT-4.1 in the API. OpenAI Technical Report (April 14, 2025), 2025a. URL https://openai.com/index/gpt-4-1/.

OpenAI. Introducing OpenAI o3 and o4-mini. OpenAI Release Notes (April 16, 2025), 2025b. URL https://openai.com/index/introducing-o3-and-o4-mini/.

Prakash, N., Shaham, T. R., Haklay, T., Belinkov, Y., and Bau, D. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *Proceedings of the 2024 International Conference on Learning Representations*, 2024. arXiv:2402.14811.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sharma, P., Rott Shaham, T., Baradad, M., Fu, S., Rodriguez-Munoz, A., Duggal, S., Isola, P., and Torralba, A. A vision check-up for language models. In *arXiv preprint*, 2024.

Shirai, K., Hashimoto, A., Nishimura, T., Kameko, H., Kurita, S., Ushiku, Y., and Mori, S. Visual recipe flow: A dataset for learning visual state changes of objects with recipe flows. In Calzolari, N., Huang, C.-R., Kim, H., Pustejovsky, J., Wanner, L., Choi, K.-S., Ryu, P.-M., Chen, H.-H., Donatelli, L., Ji, H., Kurohashi, S., Paggio, P., Xue, N., Kim, S., Hahm, Y., He, Z., Lee, T. K., Santus, E., Bond, F., and Na, S.-H. (eds.), *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 3570–3577, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.315/.

Tandon, N., Sakaguchi, K., Dalvi, B., Rajagopal, D., Clark, P., Guerquin, M., Richardson, K., and Hovy, E. A dataset for tracking entities in open domain procedural text. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6408–6417, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.520. URL https://aclanthology.org/2020.emnlp-main.520/.

Toshniwal, S., Wiseman, S., Livescu, K., and Gimpel, K. Chess as a testbed for language model state tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11385–11393, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Yoneda, T., Fang, J., Li, P., Zhang, H., Jiang, T., Lin, S., Picker, B., Yunis, D., Mei, H., and Walter, M. R. Statler: State-maintaining language models for embodied reasoning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15083–15091. IEEE, 2024.

A. Appendix

A.1. Models Struggle to Integrate Mixed Modalities

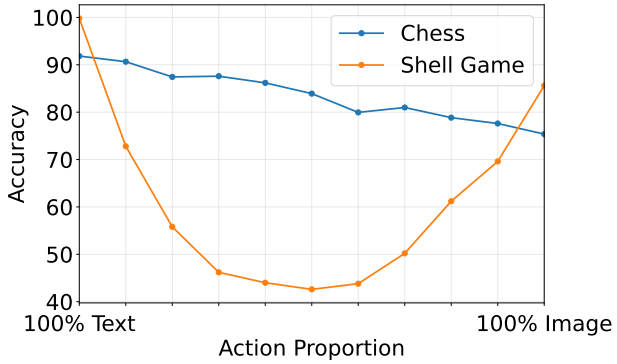


Figure 5: Chain-of-thought entity tracking accuracy for Chess and Shell Game with GPT-4o. The data splits range from 100% text-encoded actions to 100% image-encoded actions. The plot illustrates the change in accuracy as actions shift between modalities.

To examine how well models can integrate mixed-modality information, we evaluate performance as we vary the proportion of text and image-based action representations. As shown in Figure 5, for Chess performance degrades smoothly as the fraction of image actions increases, rather than exhibiting an abrupt collapse. However for Shell Game, the opposite is true and mixes of text and image actions are challenging for the model to reason over.

A.2. Fine-Tuning Improves Multimodal Entity Tracking

FINE-TUNED	TEXT	IMAGE
CHESS		
GPT-4O MINI	89.2	-
GPT-4O	97.0	86.4
SHELL GAME (S=3)		
GPT-4O MINI	32.0	-
GPT-4O	74.0	32.0

Table 4: Fine-tuned model entity tracking accuracy for the Chess and Shell Game domains (Text actions and Image actions).GPT-4o-mini does not support image finetuning. A training set of 100 action sequences of length ten were used for Chess and 20 action sequences of length three and five for Shell Game.

Fine-tuning using the OpenAI fine-tuning API substantially improves model performance across both text and image modalities, as shown in Table 4. In Chess, fine-tuned models outperform even the strongest zero-shot reasoning models, achieving 97.0% accuracy in the text domain and a significant boost to 86.4% in the image domain. This suggests that even with a relatively small dataset, fine-tuning allows the model to learn entity tracking representations that generalize better in both modalities. Notably, fine-tuning leads to a larger improvement in the image modality than in the text modality. This reinforces the idea that pretrained models already encode strong textual reasoning capabilities, whereas multimodal reasoning requires additional adaptation.

In contrast, improvements on a simplified version of Shell Game with only three moves are minimal in case of image-encoded actions. The Shell Game task is not present in the training data and it's harder for the model to generalize, even when exposed to a large fraction of the possible games of the given length. This may indicate that the Shell Game domain is simply too challenging for the model to learn in both the image and text settings from a limited number of examples. A more complex training curriculum involving fine-tuning over multiple game lengths may be required.

Model Name	Image	Reasoning
Claude 3.7 Sonnet (Anthropic, 2025)	✓	
Claude 3.7 Sonnet Thinking (Anthropic, 2025)	✓	✓
Gemini-2.5-Flash (DeepMind, 2024)	✓	
Gemini-2.5-Flash Reasoning (DeepMind, 2024)	✓	✓
Gemini-2.5-Pro (DeepMind, 2024)	✓	✓
GPT-4o (OpenAI, 2024b)	✓	
GPT-4o mini (OpenAI, 2024a)	✓	
GPT-4.1 (OpenAI, 2025a)	✓	
GPT-4.1-mini (OpenAI, 2025a)	✓	
GPT-4.1-nano (OpenAI, 2025a)	✓	
Llama 4 Mavrick (Meta, 2025)	✓	
Minimax-VL-01 (MiniMax et al., 2025)	✓	
o1 (OpenAI, 2024c)	✓	✓
o3 (OpenAI, 2025b)	✓	✓
o4-mini (OpenAI, 2025b)	✓	✓

Table 5: Comparison of capabilities of language models evaluated using the MET benchmark. All evaluated models support text input and output. The total API cost of experiments run is $2500.00.

CONFIGURATION	EPOCHS	LRM	BATCH
CHESS			
TEXT:			
GPT-4O-MINI	3	1.8	1
GPT-4O	3	2	1
IMAGE:			
GPT-4O	3	2	1
SHELL GAME			
TEXT:			
GPT-4O-MINI	3	1.8	1
GPT-4O	3	2	1
IMAGE:			
GPT-4O	5	2	1

Table 6: Hyperparameters used for fine-tuning across domains and modalities. The training epochs, learning rate multiplier (LRM), and batch size are reported.

Role	Messages

User	`The shell game is a classic game where a ball is hidden under one of three shells. You are a helpful assistant that tracks the position of the ball. The ball starts under shell 2. Here are the moves played:`
	`1 swap 3`
	`2 swap 3`
	`Now what is the final position of the ball? Only output the number 1, 2, or 3.`

Assistant	`3`

Figure 6: An example zero-shot user–assistant exchange in the **Shell Game** domain, illustrating how the system tracks swaps to determine the ball's final shell.

Role	Messages

User	`You are a helpful assistant that interprets image-based actions in chess.`
	`Here is an image representing a move:`
	[Image Input]
	`In UCI notation, what move does the arrow on the chessboard represent? The move is from the green square to the red square. (e.g., 'e2e4'). Only output the move and nothing else.`

Assistant	`e2e4`

Figure 7: An example user–assistant exchange in the **Chess** domain, where the assistant identifies the move represented in the image.

Role	Messages

User	`You are a helpful assistant that interprets image-based actions in the shell game.`
	`Here is an image representing a swap:`
	[Image Input]
	`In shell game notation, which shells are being swapped in the image? Shells are labeled '1', '2', '3' and the shells being swapped have their numbers highlighted in green. Only output a dash-separated pair like '1 swap 3' and nothing else.`

Assistant	`1 swap 3`

Figure 8: An example user–assistant exchange in the **Shell Game** domain, where the assistant identifies the shell swap represented in the image.

A.3. Models

The models evaluated using MET-Bench are listed in Table 5.

Llama 4 Maverick This model is released under the license: https://github.com/meta-llama/llama-models/blob/main/models/llama4/LICENSE. The model is 400 billion parameters and is trained on a " 22 trillion tokens of multimodal data from a mix of publicly available, licensed data and information from Meta's products and services" (Meta, 2025).

Minimax-VL-01 This model is released under the license: https://github.com/MiniMax-AI/MiniMax-01/blob/main/LICENSE. The model is 465 billion parameters and is trained on a "diverse [dataset] incorporating diverse sources including academic literature, books, web content, and programming code" and post-training dataset encompassing many multimodal and NLP tasks of 512 billion tokens (MiniMax et al., 2025).

A.3.1. PROPRIETARY MODELS

These models have limited information about their training and development. Like Minimax-VL-01, these models are likely trained on diverse, web-scale corpora spanning many domains and tasks. We provide links to the current terms of their use.

Claude 3.7 Sonnet https://www.anthropic.com/legal/consumer-terms.

Gemini-2.5 https://ai.google.dev/gemini-api/terms

GPT-4o mini, GPT-4o, GPT-4.1, GPT-4.1-mini, GPT-4.1-nano, o4-mini, o3 https://openai.com/policies/

A.4. Datasets

The Chess dataset is adapted from Toshniwal et al. (2022) which is adapted from the MillionBase dataset, available for download at https://rebel13.nl/rebel13/rebel%2013.html. To the best of our knowledge, no license or terms of use are currently listed for either the original MillionBase dataset or dataset of Toshniwal et al. (2022). Our usage of this dataset is consistent with the description of its use by Toshniwal et al. (2022).

MET-Bench is intended for evaluating and improving the ability of VLMs to perform entity tracking. It is released under the MIT License.