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

Anonymous ACL submission

Abstract

Entity tracking is a fundamental challenge in natural language understanding, requiring models to maintain coherent representations of entities. Previous work has benchmarked entity tracking performance in purely text-based tasks. 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 effectively current models integrate textual and image-based state updates. Our findings reveal a significant performance gap between text-based and image-based tracking and that this performance gap stems from deficits in visual reasoning rather than perception. We further show that explicit text-based reasoning strategies improve performance, yet substantial limitations remain, especially in long-horizon multimodal scenarios. Our results highlight the need for improved multimodal representations and reasoning techniques to bridge the gap between textual and visual entity tracking.

1 Introduction

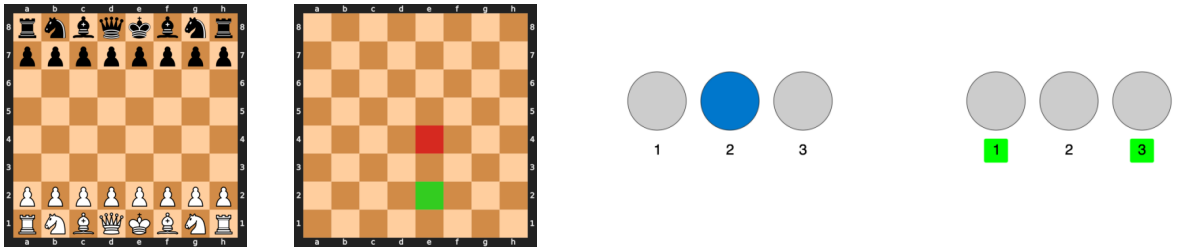
Natural language understanding requires the ability to track and update information about entities as they evolve through text. From coreference resolution (Hobbs, 1978; Lappin and Leass, 1994) and discourse processing to narrative comprehension, computational linguistics has long grappled with the challenge of maintaining coherent entity representations across textual contexts (Bunescu and Paşca, 2006; Schank and Abelson, 1977).

While significant progress has been made in tasks like coreference resolution and entity linking (Liu et al., 2023; Papalampidi et al., 2022), the broader challenge of tracking entity states—understanding how entities change through sequences of actions or events—remains an open challenge (Fagnou et al., 2024; Kim and Schuster, 2023; Toshniwal et al., 2022). This limitation becomes particularly apparent in tasks requiring integration of information across multiple modalities, an increasingly important frontier in computational linguistics as language processing systems are asked to reason about content that combines text with other forms of communication like images and video.

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 like discourse processing and situated language understanding, while also connecting to emerging research in multimodal dialogue and human-AI interaction. We focus specifically on scenarios where language models must reason about world-state changes described through a combination of text and images. Consider the task of understanding assembly instructions that combine written steps with supporting diagrams: while text might specify "Attach panel A to the base using the provided screws," accompanying images show the precise alignment and orientation. Accurate language understanding in such contexts requires maintaining a coherent mental model that integrates both linguistic and visual information about how entities' states evolve.

To systematically evaluate models' capabilities in this multimodal language understanding setting, we introduce two complementary benchmarks: *multimodal Chess* and the *Shell Game*. Through these domains, we assess how effectively current language models can track entity states when updates are conveyed through both text and images. Our analysis reveals substantial disparities in how models process text-based and image-based entity-state updates, highlighting fundamental limitations in their multimodal language understanding.

While Chess and the Shell Game provide structured testbeds for evaluating entity tracking, real-



(a) Chess State (b) Chess Action (c) Shell Game State (d) Shell Game Action

Figure 1: Illustration of the two domains used in this work. (a) An example Chess board state. (b) The chess move (action) rendered as an image. (c) An initial shell game state with the blue ball under a shell. (d) Image action representing shells at positions one and three being swapped.

world applications often involve more ambiguous and dynamic environments. However, by isolating state-tracking performance in controlled settings, we establish a clear baseline for assessing multimodal reasoning, one that provides a straightforward means of evaluation and can scale in difficulty with minimal changes.

We hypothesize that current language models struggle with multimodal entity tracking not due to low-level perceptual failures but because they lack representations (learned or otherwise engineered) for maintaining entity coherence 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 probing 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 traditional NLP methods like fine-tuning when generalizing to novel domains.

2 Background

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. Formally, at each time step t, the environment is in state $\boldsymbol{S_t}$, and transitions occur according to an action sequence $\boldsymbol{A} = (\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_T})$. The agent receives observations $\boldsymbol{A_t}$ corresponding to each action, which may be textual ($\boldsymbol{A_t}^{\text{text}}$) or visual ($\boldsymbol{A_t}^{\text{image}}$). The objective is to infer the final state,

$$\boldsymbol{S_T} = f(\boldsymbol{S_0}, \boldsymbol{A_1}, \boldsymbol{A_2}, \ldots, \boldsymbol{A_T}),$$

where f is the function modeling entity state updates.

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 while processing state transitions visually. By doing so, we assess their capacity to maintain coherent entity representations across modalities while minimizing confounding errors from perceptual failures, which remain a known limitation of current vision-language models (Sharma et al., 2024).

2.1 Chess Domain

Chess is a well-studied domain for testing entity tracking of deep learning models (Toshniwal et al., 2022). The state S_t represents 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 notation) or visual (board images) descriptions of moves. We likewise

adopt chess as an entity tracking testbed where the task is to maintain a correct representation of the board state across a sequence of moves. This distinction foregrounds how well a model can integrate and track piece locations as they change over time in potentially complicated board configurations.

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.

- **Text-Encoded Moves**: Each action is provided as a short UCI textual description (e.g., "`e2e4`" for moving a piece from e2 to e4).
- **Image-Encoded Moves**: Each action is accompanied by a rendered image that serves as a visual representation of the move (see Fig. 1).

In both cases, the final output is the FEN-encoded location of each piece on the board after a sequence of moves. The dataset includes multiple game trajectories of varying length, capturing a variety of piece types and board states.

2.2 Shell Game Domain

The second domain in MET-Bench is the Shell Game, a 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 S_t tracks the hidden position of a ball under three shells, 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 and Schuster, 2023).

This domain has a simpler entity-state and action space than Chess. However, while many frontier models have been trained on UCI/FEN encoded chess games, the Shell Game is, to the best of our knowledge, not present in the training data of these models. There may however be analogous tasks in the pre-training data.

We simulate repeated Shell Game swaps to create a set of Shell Game trajectories. Swap actions are either:

- **Text-Encoded Swap**: Denoted as "`x swap y`", where $\{x, y\}$ are in $\{1, 2, 3\}$.
- **Image-Encoded Swap**: An image depicting the shells being swapped, with the ball visually hidden (see Fig. 1).

The ground truth entity state after the game finishes is a single number indicating the final shell position of the ball.

2.3 Models

We use MET-Bench to evaluate the limitations of frontier models, including vision-language models (VLMs) which accept images and text as input, and newer reasoning models like OpenAI's o1 that are trained using reinforcement learning and utilize test-time search algorithms to improve their reasoning abilities on domains like mathematics and coding. A full list of models and their capabilities is shown in Appendix A, Table 5.

3 Methods

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? Only output the FEN.`

Assistant	`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.

We utilize the standard chat-based schema exposed by current frontier models that consists of interleaved user-provided and assistant (model) provided messages. Figures 2 shows the prompting strategy used for the Chess domain. Similarly, Appendix A, 6 shows the prompting strategy used for the Shell Game domain. Text actions are represented using simple notation on which the models have been trained, UCI for Chess and a simple domain-specific-language for Shell Game.

In the case of image-action input, the text actions are replaced with their image-rendered versions as

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

Base64 encoded PNG images and a text description of how to interpret the image-actions is provided. Fig. 1 shows the image representations used for the Chess and Shell Game actions. These image representations were created through visual-prompt engineering to maximize the classification accuracy of actions depicted. Various common image representations were explored including arrows, bounding boxes, and symbolic markers. The image depiction of the game actions is explained to the language model every time images are provided using the prompts in Appendix A, Figures 7 & 8.

4 Experiments

We perform experiments across a wide range of models and settings to evaluate different aspects of frontier-model entity-tracking performance. For all experiments, the models are sampled with a temperature of zero.

4.1 Tracking Entities with Text and Image Actions

We evaluate difference in accuracy when tracking images from text and image actions in the zero-shot, few-shot, and chain-of-thought settings. These results are presented in Section 5.1.

Zero-shot

Chess In the Chess domain, we evaluate on a set of 100 games selected at random from the test set, each with a sequence length of ten actions. The model must predict the FEN string for the final state. If the FEN string contains syntax errors such that it cannot be parsed, the accuracy for that instance is zero. We report the per-square accuracy of the predicted board, that is the ratio of correctly predicted pieces (or absence of a piece) to the total number of board tiles. The 'Game Start' baseline is the accuracy of predicting the initial board configuration. After only ten actions, most of the board configuration remains unchanged, so this is a strong baseline.

Shell Game We evaluate Shell Game using a set of 500 games generated at random, each with a sequence length of five actions. The final state is a single number $n \in \{1, 2, 3\}$ that gives the position of the ball, and we measure the accuracy of predicting it. The naive baseline picks a position uniformly at random.

Few-Shot and Chain-of-Thought

In these settings, the evaluation and procedure remain largely unchanged from the zero-shot Chess. For the few-shot experiments, $N = 5$ examples are selected at random from the training set and prepended to the test example. The few-shot examples have the same number of actions as the test-set examples. To evaluate the effect of chain-of-thought reasoning, we prompt the model to 'think step by step before producing a final answer.'

4.2 Sequence-Length Variation

The sequence length of the actions is varied to quantify the effect of compounding errors on the models. These sequences range from zero to 100 actions in both the image and text action modality. This serves to quantify the relative drop-off in performance among models in the zero-shot setting. These results are presented in Section 5.2.

4.3 Image-Action Classification

We perform an experiment to demonstrate that VLMs have the ability to accurately interpret the actions depicted in the image-action representations. Figures 7 & 8 in Appendix A show the prompting strategy used to get a VLM to predict (zero-shot) a text action from an image action. We perform this evaluation on a test set of 1000 image depictions of actions for each game. The results are presented in Section 5.3.

4.4 Cascaded Inference

Using the text actions predicted from the images in the image-action classification task, we devise an ablation to test the effect of cascading (performing multimodal inference in two steps) in the zero-shot setting. Given a sequence of image-based observations $\boldsymbol{A}_t^{\text{image}}$, a vision-language model (VLM) first predicts the corresponding text-based action sequence $\hat{\boldsymbol{A}}^{\text{text}} = g(\boldsymbol{A}^{\text{image}})$, where g maps images to text using the procedure in Section 4.3. The text actions are then used for zero-shot inference to estimate the final state as $\boldsymbol{S_T} = f(\boldsymbol{S_0}, \hat{\boldsymbol{A}}^{\text{text}})$. This removes the need for the model to perform entity tracking directly in the image modality, isolating the effect of perceptual failures. These results are presented in Section 5.4.

4.5 Fine-Tuning Experiments

We fine-tune frontier OpenAI models using their fine-tuning API on a small training set. We evaluate these in the Chess and Shell Game domains using

Chess	Text	Image
Baseline		
Game Start	74.4	74.4
Zero-Shot		
Claude 3.5 Sonnet	96.8	66.2
MiniMax-VL-01	86.3	65.8
Gemini-2.0-Flash	93.3	74.7
GPT-4o mini	68.3	60.6
GPT-4o	89.6	73.3
Few-Shot (n=5)		
Claude 3.5 Sonnet	99.6	77.9
MiniMax-VL-01	88.0	75.9
Gemini-2.0-Flash	96.6	78.7
GPT-4o mini	75.1	74.9
GPT-4o	94.9	77.7
Chain-of-Thought		
Claude 3.5 Sonnet	97.9	68.4
MiniMax-VL-01	62.7	58.0
Gemini-2.0-Flash	93.3	74.5
GPT-4o mini	72.0	69.6
GPT-4o	91.8	74.9
Reasoning		
o1-mini	65.1	-
o3-mini	99.6	-
o1	98.2	83.5

(a) Chess with ten moves in each sequence.

Shell Game	Text	Image
Baseline		
Random	33.3	33.3
Zero-Shot		
Claude 3.5 Sonnet	34.4	36.2
MiniMax-VL-01	34.4	35.2
Gemini-2.0-Flash	30.0	33.4
GPT-4o mini	30.6	32.8
GPT-4o	36.0	32.4
Few-Shot (n=5)		
Claude 3.5 Sonnet	34.0	30.6
MiniMax-VL-01	36.4	32.0
Gemini-2.0-Flash	37.0	31.4
GPT-4o mini	34.4	31.2
GPT-4o	37.2	31.0
Chain-of-Thought		
Claude 3.5 Sonnet	97.4	94.2
MiniMax-VL-01	92.6	32.8
Gemini-2.0-Flash	76.8	33.8
GPT-4o mini	84.4	35.0
GPT-4o	99.8	84.2
Reasoning		
o1-mini	99.8	-
o3-mini	100.0	-
o1	100.0	92.8

(b) Shell Game with five swap moves in each sequence.

Table 1: Entity tracking accuracy in Chess and Shell Game for text and image actions. In the Few-Shot setting $N = 5$ in-context examples are used. Methods and models which employ explicit reasoning perform best (chain-of-thought and reasoning models).

both image and text modalities. For Chess, we select a random set of 100 training examples of sequence length ten. For Shell Game we use 20 training examples of sequence length 3. Results are available in Section A.2. Hyperparameters were determined through grid search and are available in Appendix A.2, Table 6.

4.6 Mixed-Modality Experiments

We perform an ablation where the image-action and text-action modalities are mixed. At each action step, one action modality is randomly selected with a probability varying from 100% text actions to 100% image actions. This serves to quantify the effect of forcing the model to reason over multiple input modalities simultaneously. These results are presented in Section A.1.

5 Results

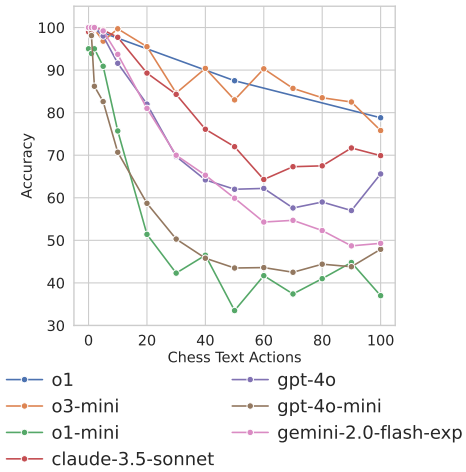
5.1 Tracking in Text Outperforms Images

Chess Table 1a reports the accuracy for text and image actions in the chess domain. Performance in the text modality is significantly better than in the image modality. In the zero-shot setting, the best-performing text model, Claude 3.5 Sonnet, achieves an impressive 96.8% accuracy, while its image-based counterpart drops sharply to 66.2%. A similar trend holds across models, indicating that current models can perform entity tracking in text but fail to integrate visual updates as effectively.

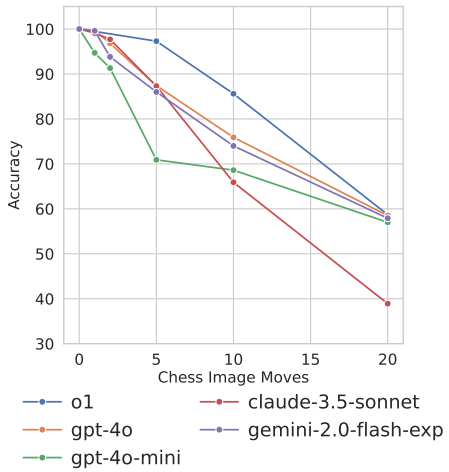
Both in-context learning with few-shot prompting and chain-of-thought reasoning lead to performance improvements. In the image modality, the accuracy is not significantly greater than the naive baseline. Only the o1 reasoning model achieves an accuracy greater than 80% in the image modality.

Shell Game Table 1b reports the accuracy for text and image actions for Shell Game. The results follow a pattern similar to that of Chess, with text-based tracking outperforming image-based tracking. However, unlike in Chess, the performance in the zero-shot setting is close to random. But the best model, o1, attains accuracies of 100.0% and 92.8% for the text and image modalities, respectively.

Few-shot prompting provides only marginal im-



(a) Chess zero-shot accuracy with text actions.



(b) Chess zero-shot accuracy with image actions.

Figure 3: In the text action setting, the reasoning models, o1 and o3-mini, maintain the highest accuracy at longer sequence lengths. o1-mini and the other models begin to output invalid and inaccurate board representations. All models struggle to maintain accurate board representations in the image action setting, with o1 performing the best.

provements, but chain-of-thought gives large performance increases. GPT-4o's accuracy jumps from 36.0% to 99.8% in text, and from 32.4% to 84.2% in image tracking. These results suggest that when guided to decompose the task step-by-step, models can reason more effectively using image inputs, a finding that complements the results of performing cascaded inference in Section 5.4.

5.2 Reasoning Aids Long Sequence Accuracy

Chess Figure 3a plots model accuracy against increasing sequence lengths of text actions. The models are evaluated in the zero-shot setting for sequence lengths of zero to 100 text actions. Reasoning models like o1 and o3-mini are able to handle longer sequence lengths with a smaller decrease in accuracy. However, o1-mini performs worse than the other models as it produces more invalid FEN board representations at longer sequence lengths. In contrast, the non-reasoning models experience sharp decreases in accuracy after only a few actions. Figure 3b plots model accuracy against increasing sequence lengths of image actions. The models are evaluated in the zero-shot setting for sequence lengths of zero to 20 image actions. While the reasoning models attain higher accuracies, the performance differences are smaller than in the text-action setting.

Shell Game In the text modality in Figure 4a, the reasoning models o1, o1-mini, and o3-mini attain the highest accuracies. o1 performs perfectly at a sequence length of 50 actions, where the non-reasoning models' performance is significantly degraded. In the image modality results in Figure 4b, o1 performs better than the other models, but sees a rapid decrease in accuracy with sequence lengths longer than five actions. By 20 actions, the performance of all models has degraded to random.

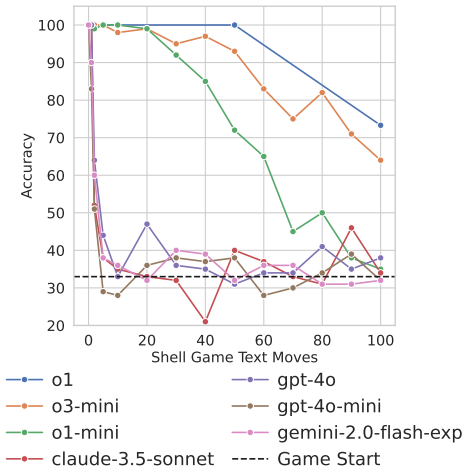
Model	Start	End	Overall
Chess			
GPT-4o-mini	62.1	55.2	41.0
GPT-4o	97.3	97.0	94.5
Shell Game			
GPT-4o-mini	100.0	100.0	100.0
GPT-4o	100.0	100.0	100.0

Table 2: Percent image-action classification accuracy for various models. We report the accuracy of predicting the action start, end, and overall/UCI action for both Chess and Shell Game on 1,000 image actions.

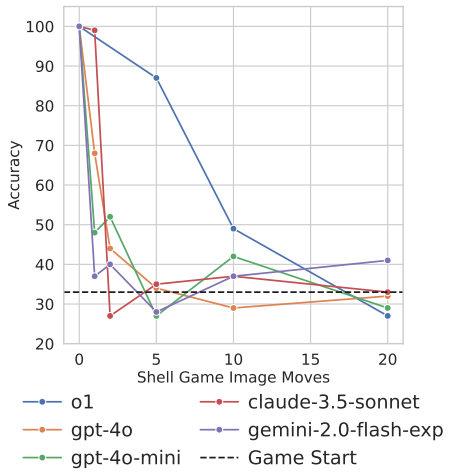
The superior performance of reasoning models like o1 suggests that structured inference mechanisms, such as test-time search or chain-of-thought, are crucial for maintaining coherent entity states over long sequences. This aligns with prior findings that models trained on structured reasoning tasks (e.g., mathematics, coding) develop stronger implicit state-tracking capabilities (Kim et al., 2024), whereas standard vision-language models struggle with entity persistence beyond short contexts.

5.3 Models Understand Image Actions

Table 2 shows the performance of GPT-4o and GPT-4o-mini on classifying the text action represented by each image action. For Chess, the action 'start' is the square of the piece being moved, and 'end' is



(a) Shell Game zero-shot accuracy with text actions.



(b) Shell Game zero-shot accuracy with image actions.

Figure 4: In the text action setting, the reasoning models o1, o1-mini, and o3-mini achieve the highest performance over long action sequences, but performance degrades for all models as sequence length increases. In the image action setting, o1 performs the best, but achieves an accuracy no better than guessing the starting state by 20 actions.

	Cascaded	
Method	**Chess**	**Shell**
Baseline		
Game Start	74.4	33.3
Zero-Shot		
GPT-4o mini	63.0	28.0
GPT-4o	89.8	34.0

Table 3: Cascaded entity tracking accuracy 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.

the destination square. For Shell Game, the 'start' is the first shell to be swapped and 'end' is the second. 'Overall' is the accuracy of classifying the entire action (start and end) correctly. While GPT-4o-mini struggles to recognize actions in chess, GPT-4o achieves an accuracy of 94.5%. Both 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.

A potential concern is whether the observed failures stem from poor image representation rather than reasoning deficiencies. However, our cascaded inference experiments (Section 5.4) demonstrate that when models first translate image actions into text, they achieve near-text-level accuracy. This suggests that models can correctly parse image-based actions but struggle with integrating them into coherent state updates, a limitation in reasoning rather than perception.

5.4 Cascading Matches Text-Only Tracking

Table 3 shows the accuracy of cascaded inference in the zero-shot setting for GPT-4o and GPT-4o-mini. In this setting, the image actions are first translated into text actions, and then run through the text-based entity tracking pipeline. 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.

6 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 GPT-4o's accuracy from 36.0% to 99.8% for text and 32.4% to 84.2% 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 o1 and o3-mini 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.

7 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 and 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 and 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.

8 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.

9 Limitations

Our benchmark is restricted to two synthetic domains—Chess and the Shell Game. While these domains are well-structured and offer a clear framework for assessing entity state tracking, they may not fully capture the complexity of real-world multimodal reasoning tasks that require entity tracking.

Our study highlights a substantial performance gap between text-based and image-based entity tracking, yet the exact causes of this disparity remain unclear. While our cascaded inference experiments suggest that models struggle to reason directly over visual updates rather than simply perceiving them, further investigation is needed to pinpoint the underlying source of the problem and whether it's possible to explicitly ameliorate it.

Additionally, our fine-tuning experiments demonstrate that performance can be improved with domain-specific adaptation, but we do not investigate the trade-offs between fine-tuning and generalization across unseen multimodal datasets. Understanding whether these improvements transfer to novel multimodal reasoning tasks is an important avenue for future work.

Ethical Considerations

Vision-language models may inherit biases from their training data or other aspects of their development, leading to disparities in performance across different languages, cultures, or other categories that intersect with textual and visual representations. The evaluations in this work can be used to improve the capabilities of frontier models, which if deployed in contexts requiring multimodal entity tracking, could further expose end-users to the biases present in this work including our choice of game domains and language.

Acknowledgments

This paper and associated code were created with the assistance of ChatGPT for proofreading and debugging.

References

Anthropic. 2024. Claude 3.5 Sonnet. Anthropic Announcement (June 21, 2024).

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16, Trento, Italy. Association for Computational Linguistics.

Erwan Fagnou, Paul Caillon, Blaise Delattre, and Alexandre Allauzen. 2024. Chain and causal attention for efficient entity tracking. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13174–13188, Miami, Florida, USA. Association for Computational Linguistics.

Aditya Gupta and Greg Durrett. 2019. Effective use of transformer networks for entity tracking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 759–769, Hong Kong, China. Association for Computational Linguistics.

Demis Hassabis and Koray Kavukcuoglu. 2024. Introducing Gemini 2.0: our new AI model for the agentic era. Google DeepMind Blog (Dec. 11, 2024).

Jerry R. Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Najoung Kim, Sebastian Schuster, and Shubham Toshniwal. 2024. Code pretraining improves entity tracking abilities of language models. *arXiv preprint*.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *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)*, pages 1813–1827, Online. Association for Computational Linguistics.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*.

Ruicheng Liu, Rui Mao, Anh Tuan Luu, and Erik Cambria. 2023. A brief survey on recent advances in coreference resolution. *Artificial Intelligence Review*, 56(12):14439–14481.

Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. In *Proceedings of the 54th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1456–1465, Berlin, Germany. Association for Computational Linguistics.

MiniMax, Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, Enwei Jiao, Gengxin Li, Guojun Zhang, Haohai Sun, Houze Dong, Jiadai Zhu, Jiaqi Zhuang, Jiayuan Song, Jin Zhu, Jingtao Han, Jingyang Li, Junbin Xie, Junhao Xu, Junjie Yan, Kaishun Zhang, Kecheng Xiao, Kexi Kang, Le Han, Leyang Wang, Lianfei Yu, Liheng Feng, Lin Zheng, Linbo Chai, Long Xing, Meizhi Ju, Mingyuan Chi, Mozhi Zhang, Peikai Huang, Pengcheng Niu, Pengfei Li, Pengyu Zhao, Qi Yang, Qidi Xu, Qiexiang Wang, Qin Wang, Qiuhui Li, Ruitao Leng, Shengmin Shi, Shuqi Yu, Sichen Li, Songquan Zhu, Tao Huang, Tianrun Liang, Weigao Sun, Weixuan Sun, Weiyu Cheng, Wenkai Li, Xiangjun Song, Xiao Su, Xiaodong Han, Xinjie Zhang, Xinzhu Hou, Xu Min, Xun Zou, Xuyang Shen, Yan Gong, Yingjie Zhu, Yipeng Zhou, Yiran Zhong, Yongyi Hu, Yuanxiang Fan, Yue Yu, Yufeng Yang, Yuhao Li, Yunan Huang, Yunji Li, Yunpeng Huang, Yunzhi Xu, Yuxin Mao, Zehan Li, Zekang Li, Zewei Tao, Zewen Ying, Zhaoyang Cong, Zhen Qin, Zhenhua Fan, Zhihang Yu, Zhuo Jiang, and Zijia Wu. 2025. Minimax-01: Scaling foundation models with lightning attention. *Preprint*, arXiv:2501.08313.

OpenAI. 2024a. GPT-4o mini: advancing cost-efficient intelligence. OpenAI Blog (July 18, 2024).

OpenAI. 2024b. Hello GPT-4o. OpenAI Announcement (May 13, 2024).

OpenAI. 2024c. Introducing OpenAI o1-preview and o1-mini. OpenAI Release Notes (Sept. 12, 2024).

OpenAI. 2024d. Introducing OpenAI o1-preview and o1-mini. OpenAI Release Notes (Sept. 12, 2024).

OpenAI. 2025. OpenAI o3-mini System Card. OpenAI Technical Report (Jan. 31, 2025).

Pinelopi Papalampidi, Kris Cao, and Tomas Kocisky. 2022. Towards coherent and consistent use of entities in narrative generation. In *International Conference on Machine Learning*, pages 17278–17294. PMLR.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *Proceedings of the 2024 International Conference on Learning Representations*. ArXiv:2402.14811.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. 2024. A vision check-up for language models. In *arXiv preprint*.

Keisuke Shirai, Atsushi Hashimoto, Taichi Nishimura, Hirotaka Kameko, Shuhei Kurita, Yoshitaka Ushiku, and Shinsuke Mori. 2022. Visual recipe flow: A dataset for learning visual state changes of objects with recipe flows. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3570–3577, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Takuma Yoneda, Jiading Fang, Peng Li, Huanyu Zhang, Tianchong Jiang, Shengjie Lin, Ben Picker, David Yunis, Hongyuan Mei, and Matthew R Walter. 2024. Statler: State-maintaining language models for embodied reasoning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15083–15091. IEEE.

A Appendix

A.1 Models Struggle to Integrate Mixed Modalities

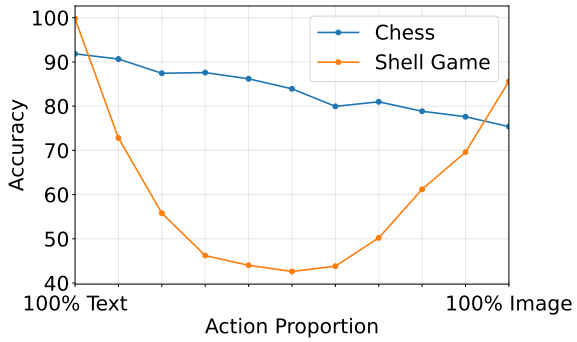


Figure 5: Chain-of-thought entity tracking accuracy for Chess and Shell Game with GPT-4o. The data splits range from 100% text-encoded actions to 100% image-encoded actions. The plot illustrates the change in accuracy as actions shift between modalities.

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 3.5 Sonnet (Anthropic, 2024)	✓	
Gemini-2.0-Flash (Hassabis and Kavukcuoglu, 2024)	✓	
GPT-4o mini (OpenAI, 2024a)	✓	
GPT-4o (OpenAI, 2024b)	✓	
Minimax-VL-01 (MiniMax et al., 2025)	✓	
o1-mini (OpenAI, 2024c)		✓
o1 (OpenAI, 2024d)	✓	✓
o3-mini (OpenAI, 2025)		✓

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 $2340.00.

A.3 Models

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

Minimax-VL-01 This model is released under the license: `https://github.com/MiniMax-AI/MiniMax-01/blob/main/LICENSE`. The model is

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.

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

Gemini-2.0-Flash https://ai.google.dev/gemini-api/terms

GPT-4o mini, GPT-4o, o1, o1-mini, o3-mini 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.