The Representation-Reasoning Gap: Language Models Encode Spatial Relations They Fail to Reason Over

Jefrey Bergl*
Department of Computer Science
University of North Carolina at Chapel Hill
Independent Research

Abstract

Do language models that fail at spatial reasoning nonetheless encode spatial information in their internal representations? We investigate this question by testing three autoregressive models (GPT-2 Medium, Pythia 1.4B, Pythia 2.8B) on a synthesized grid-world navigation task requiring 1–5 sequential movement steps. Linear probes (ridge classifiers) trained on residual stream activations recover the agent's grid position with 98–100% accuracy after a single step, while behavioral accuracy on the same inputs is 0.0–5.2%, yielding a representation-reasoning gap of $\Delta = 94.8$–100.0 percentage points. The gap persists at all step counts and does not close with model scale across the 355M–2.8B range. A persistence analysis reveals that position representations *consolidate* at the final token, where final-token probe accuracy exceeds step-token accuracy by up to 22 percentage points at N=5, yet the models still produce incorrect outputs. These results demonstrate that representation formation and reasoning over representations are dissociable processes: spatial position is linearly decodable from activations that the model cannot convert into correct behavior.

1 Introduction

Recent work has established that language models form internal representations with surprising geometric structure. Gurnee and Tegmark [5] demonstrated that spatial and temporal coordinates are linearly decodable from the residual stream of LLaMA-2 models, and Li et al. [7] showed that a transformer trained on Othello move sequences develops a linear representation of the board state. These findings support the linear representation hypothesis [9]: high-level concepts are encoded as directions in activation space. Yet a persistent puzzle remains: if models encode spatial structure, why do they fail at spatial reasoning tasks that require composing multiple spatial updates?

This paper tests a specific version of this puzzle. We construct a minimal spatial reasoning task, navigating an agent on a 5×5 grid through N sequential directional moves, and measure both the *behavioral* accuracy (does the model produce the correct final position?) and the *representational* accuracy (can a linear probe decode the correct position from the model's activations at each step?). If both degrade together, representations and reasoning are coupled. If probe accuracy remains high while behavioral accuracy collapses, a dissociation exists: the model encodes position but cannot reason over it.

*Correspondence: jbergl@unc.edu

We find a large dissociation. Across three models spanning 355M to 2.8B parameters, linear probes recover the agent's current position with 98–100% accuracy at N=1, while behavioral accuracy is 0.0–5.2% on the same inputs. The gap narrows at higher step counts as probe accuracy degrades, but the probe consistently exceeds behavioral performance. Representation persistence analysis reveals that position information does not merely fade before reaching the output. Final-token probes actually *outperform* step-token probes at $N \geq 3$, indicating that the model consolidates spatial information at the answer position but fails to convert it into a correct output. Intermediate tracking shows that the starting position persists strongly across the sequence (100% $\rightarrow$ 36–73% by step 5) while the current position degrades more rapidly (100% $\rightarrow$ 13–17%), suggesting models anchor to the initial state rather than maintaining a running spatial update.

This work contributes a controlled experimental demonstration that representation formation and compositional reasoning are dissociable in language models, quantified across three model scales. The task design isolates spatial composition from other confounds, and the probing analysis traces the representation-reasoning gap from individual movement steps through to the final output token. The results complement prior findings on the compositionality gap [10] and the limitations of transformers on compositional tasks [3], grounding these observations in the geometry of internal representations.

2 Related Work

World models in language models. The question of whether language models build internal world models has received substantial recent attention. Gurnee and Tegmark [5] probed LLaMA-2 models and found that spatial coordinates (latitude, longitude) and temporal information (year) are linearly decodable from residual stream activations, establishing that large language models encode structured spatial information. Li et al. [7] demonstrated that a GPT variant trained on Othello move sequences develops a linear representation of the 8×8 board state, despite never observing board positions during training. Nanda et al. [8] reproduced and extended this result, confirming the linearity of the emergent world model. Park et al. [9] formalized the linear representation hypothesis, the claim that concepts correspond to directions in activation space, and provided theoretical conditions under which it holds. Engels et al. [4] showed that not all features are linear, identifying multi-dimensional and circular representations, but the linear case remains the dominant paradigm for spatial and categorical features. Our work differs from these prior studies in a critical respect: rather than demonstrating that representations *exist*, we test whether they are *used* for downstream reasoning.

Compositionality and multi-step reasoning failures. Press et al. [10] introduced the compositionality gap, defined as the fraction of multi-hop questions a model answers incorrectly despite answering all constituent sub-questions correctly, and showed it remains substantial even in large models. Dziri et al. [3] provided a theoretical and empirical analysis of transformer limitations on compositional tasks, demonstrating that transformers trained on multi-step reasoning tend to linearize the computation rather than performing true composition. Our grid-world task is a clean instance of sequential composition: each step requires updating a position based on a direction, and the final answer requires composing all updates. The near-zero behavioral accuracy we observe at N=1 goes beyond a compositionality gap, since these models fail even at the single-step case, suggesting the bottleneck is in output generation rather than multi-step composition alone.

Probing methodology. Belinkov [1] surveyed the probing paradigm, noting that high probe accuracy demonstrates linear *accessibility* of information but does not establish that the model *uses* that information during inference. Hewitt and Liang [6] proposed control tasks to distinguish genuine linguistic structure from memorization by the probe. We use a linear probe (ridge classifier) rather than a nonlinear one, which provides a stricter test of linear accessibility but may underestimate total information content. We do not perform causal interventions (e.g., activation patching), so our results establish a necessary but not sufficient condition for the model's possessing a spatial world model.

3 Methods

3.1 Grid-World Task Design

We define a 5×5 grid with positions labeled (r, c) where $(0, 0)$ is the top-left corner and $(4, 4)$ is the bottom-right. Four cardinal moves are defined: north ($r \leftarrow r - 1$), south ($r \leftarrow r + 1$), east ($c \leftarrow c + 1$), and west ($c \leftarrow c - 1$). A scenario consists of a random starting position sampled uniformly from the 25 cells, followed by N random valid moves, where $N \in \{1, 2, 3, 4, 5\}$. Moves that would take the agent off-grid are rejected and resampled (never clamped) to avoid introducing boundary confounds. We generated 500 scenarios per step count for a total of 2,500 scenarios, with all intermediate positions recorded for probing at each step. All scenarios were independently validated by re-simulating each movement sequence and verifying that the computed positions match the stored ground-truth labels. Starting positions are approximately uniformly distributed across the 25 cells (min 87, max 113 per cell, $\sigma = 7.1$). Final positions are non-uniform due to boundary effects: corner cells appear as final positions 49–61 times per step count versus interior cells appearing up to 37 times, because the rejection sampling of off-grid moves biases trajectories away from edges.

Each scenario is formatted as a natural language prompt that specifies the grid layout, defines the directional semantics, states the starting position, and lists each move as "Step k: The agent moves {direction}." The prompt ends with "The answer is (" to constrain the model's generation to begin with a coordinate pair. The identical prompt template is used across all models.

3.2 Behavioral Evaluation Protocol

For each scenario, we ran greedy decoding (temperature = 0.0, no sampling) and parsed the model's output for the first (x, y) coordinate pair using a regular expression. If no valid coordinate was extracted, the prediction was counted as incorrect. Behavioral accuracy was computed per step count as the fraction of 500 scenarios where the model's predicted position matched the ground-truth final position.

3.3 Activation Extraction

We used TransformerLens [8] to extract residual stream activations (post-layer-norm) at every layer during a forward pass. Activations were extracted at two types of token positions: (1) the *direction token* at each step, i.e. the token corresponding to the direction word ("north," "south," "east," or "west") in each step instruction, identified via character-to-token offset mapping to avoid matching direction words in the preamble; and (2) the *final token*, the last token before generation begins (the opening parenthesis), where the model must produce the answer. All four direction words tokenize as single tokens across all three models, and the token positions are consistent (e.g., token index 89

for step 1 in all models), which was verified by printing surrounding context tokens for representative examples at N=1, 3, and 5. Each scenario was processed individually with activations offloaded to CPU immediately via `run_with_cache(device="cpu")` to manage GPU memory.

3.4 Linear Probing

For each combination of model, layer, and step number, we trained a linear probe to classify the agent's position from the residual stream activation vector. The target label is the ground-truth position after the corresponding step, encoded as an integer 0–24 ($r \times 5 + c$) for 25-class classification. Features were standardized using `StandardScaler` (fit on training data, applied to both splits), which was critical because residual stream activations exhibit large variance differences across dimensions. We used sklearn's `RidgeClassifier` with $\alpha = 1.0$, which solves a least-squares problem in closed form and is robust to the high-dimensional regime (d_{model} up to 2,560 with only 400 training samples). An 80/20 train/test split was held constant across all (layer, step) combinations within each step count for consistency, yielding 400 training and 100 test samples per probe. Bootstrap confidence intervals (1,000 resamples, 95% CI) were computed for step-token probes; the small test set ($n = 100$) means individual probe accuracies have an estimated uncertainty of approximately ±5–8 percentage points, though this does not affect the main finding since the representation-reasoning gap at N=1 exceeds 94 percentage points. We report top-1 accuracy on the test set; top-3 accuracy (via decision function scores) is reported as a secondary metric.

3.5 Models

We evaluated three autoregressive language models spanning an order of magnitude in parameter count: GPT-2 Medium [11] (355M parameters, 24 layers, $d_{\text{model}} = 1{,}024$), Pythia 1.4B [2] (24 layers, $d_{\text{model}} = 2{,}048$), and Pythia 2.8B (32 layers, $d_{\text{model}} = 2{,}560$). All three models are publicly available without gated access and are natively supported by TransformerLens, enabling consistent activation extraction across the model family. The Pythia models provide a controlled scale comparison: they share the same architecture, training data (The Pile), and tokenizer, differing only in width and depth. Models were loaded sequentially, with each model deleted and GPU memory cleared before loading the next.

4 Experiments & Results

4.1 Behavioral Performance

Table 1 reports behavioral accuracy per step count across all three models. All models perform near chance or below on this task. At N=1, the simplest case requiring only a single position update, GPT-2 Medium and Pythia 1.4B achieve 0.0% accuracy, and Pythia 2.8B achieves 5.2% (chance level for 25 classes is 4.0%). Rather than a monotonic decline with step count, behavioral accuracy exhibits a non-monotonic odd-even pattern: N=2 and N=4 yield 16.8–28.2% accuracy across models, while N=1, 3, 5 yield 0.0–5.2%. This pattern suggests models exploit surface-level regularities in the prompt structure (e.g., the number of step lines) rather than performing spatial computation. The near-zero accuracy at N=1 is particularly striking: the task reduces to a single table lookup (starting position + one move), yet none of the models can reliably perform it. Manual inspection of model outputs confirms that the models generate syntactically valid coordinate pairs (e.g., predicting $(0, 0)$ when the ground truth is $(0, 1)$, or $(1, 1)$ when the ground truth is $(2, 1)$), confirming that the failure is in spatial computation, not output formatting.

Table 1: Behavioral accuracy, step-token probe accuracy, and final-token probe accuracy at the peak probing layer for each model. Gap is computed as step-token probe accuracy minus behavioral accuracy. All values are percentages; chance is 4.0%.

Model	Metric	N=1	N=2	N=3	N=4	N=5
GPT-2 Med. (layer 19)	Behavioral	0.0	25.2	1.2	18.2	1.8
	Probe (step)	98.0	67.0	34.0	18.0	17.0
	Probe (final)	88.0	59.0	40.0	37.0	39.0
	Δ (gap)	**+98.0**	+41.8	+32.8	−0.2	+15.2
Pythia 1.4B (layer 22)	Behavioral	0.0	28.2	0.0	18.8	0.2
	Probe (step)	100.0	79.0	41.0	21.0	14.0
	Probe (final)	100.0	80.0	50.0	38.0	32.0
	Δ (gap)	**+100.0**	+50.8	+41.0	+2.2	+13.8
Pythia 2.8B (layer 27)	Behavioral	5.2	19.0	5.2	16.8	2.6
	Probe (step)	100.0	77.0	42.0	22.0	13.0
	Probe (final)	100.0	76.0	49.0	36.0	31.0
	Δ (gap)	**+94.8**	+58.0	+36.8	+5.2	+10.4

Table 2: Top-3 probe accuracy (%) at the peak probing layer. Even when top-1 accuracy degrades at higher step counts, the correct position remains within the probe's top-3 predictions at rates far exceeding the 12.0% chance baseline (3/25).

Model	Probe type	N=1	N=2	N=3	N=4	N=5
GPT-2 Med.	Step token	100.0	76.0	43.0	36.0	38.0
	Final token	98.0	78.0	65.0	61.0	61.0
Pythia 1.4B	Step token	100.0	81.0	51.0	44.0	27.0
	Final token	100.0	85.0	61.0	52.0	55.0
Pythia 2.8B	Step token	100.0	78.0	49.0	39.0	30.0
	Final token	100.0	81.0	58.0	55.0	54.0

4.2 Linear Probes Recover Position from Internal Representations

Despite near-zero behavioral accuracy, linear probes trained on residual stream activations at the peak probing layer recover the agent's current position with high accuracy, particularly at low step counts. At N=1, probe accuracy is 98.0% for GPT-2 Medium and 100.0% for both Pythia models (Table 1). Top-3 probe accuracy at N=1 is 100.0% for all three models. Probe accuracy degrades with increasing step count, falling to 34.0–42.0% at N=3 and 13.0–17.0% at N=5, but remains well above the 4.0% chance baseline through N=5. Top-3 accuracy follows the same degradation curve but remains substantially higher: 43.0–51.0% at N=3 and 27.0–38.0% at N=5 for step-token probes, and 58.0–65.0% at N=3 and 54.0–61.0% at N=5 for final-token probes (Table 2).

4.3 Ablation A: The Representation-Reasoning Gap (Step Count Sweep)

Figure 1 visualizes the central result: the gap between probe accuracy and behavioral accuracy as a function of step count. At N=1, the gap is maximal: $\Delta = 98.0$ percentage points for GPT-2

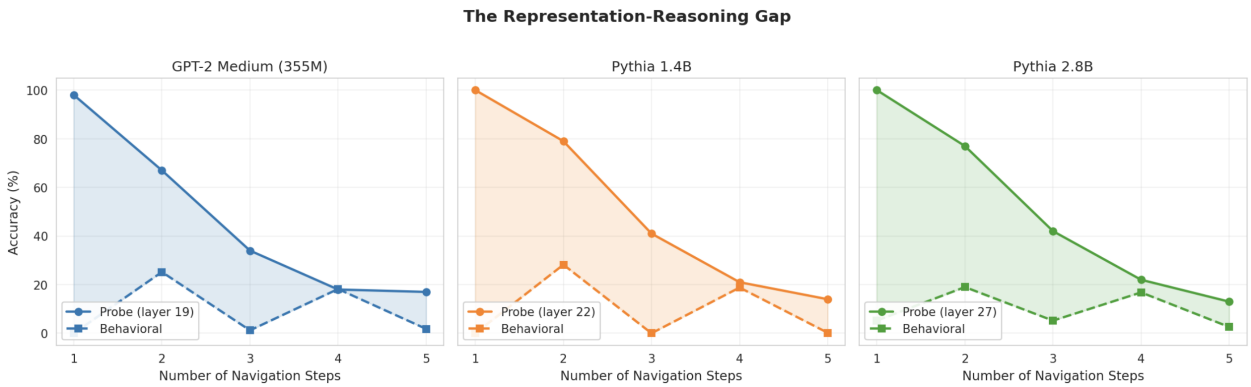

Figure 1: The representation-reasoning gap across three models. Solid lines show probe accuracy at the peak layer; dashed lines show behavioral accuracy. The shaded region highlights the gap. Probe accuracy at N=1 is 98–100% while behavioral accuracy is 0–5%. The gap does not widen monotonically due to a non-monotonic odd-even pattern in behavioral accuracy.

Medium, $\Delta = 100.0$ for Pythia 1.4B, and $\Delta = 94.8$ for Pythia 2.8B. The gap narrows at N=2 as behavioral accuracy rises on even-numbered step counts, but re-widens at N=3 and persists through N=5. At N=4, where behavioral accuracy peaks, the gap shrinks to near zero for GPT-2 Medium ($\Delta = -0.2$) but remains positive for both Pythia models ($\Delta = +2.2$ and $+5.2$). The non-monotonic pattern in behavioral accuracy means the gap does not widen monotonically with step count as originally hypothesized, but the core finding holds: the models encode position information they cannot convert to correct output.

4.4 Ablation B: Layer-by-Layer Analysis

Spatial position information is not uniformly distributed across layers. Figure 2 shows probe accuracy (step-token, last step) at every layer and step count for all three models. The pattern is consistent: early layers encode position weakly (layer 0 averages 29.0%, 33.2%, and 32.6% for GPT-2 Medium, Pythia 1.4B, and Pythia 2.8B respectively), accuracy rises through the middle layers, and peaks in the upper portion of the network. The peak probing layer is located at 79–92% of network depth: layer 19/24 for GPT-2 Medium, layer 22/24 for Pythia 1.4B, and layer 27/32 for Pythia 2.8B. Average probe accuracy at the peak layer is 46.8% (GPT-2 Medium), 51.0% (Pythia 1.4B), and 50.8% (Pythia 2.8B), compared to 29.0–33.2% at layer 0. The final layer does not show a sharp drop-off from the peak: layer 23 averages 46.4% for GPT-2 Medium and layer 31 averages 50.4% for Pythia 2.8B, indicating that spatial representations, once formed, persist through the remaining layers rather than being overwritten. This layer profile is consistent with prior findings that high-level semantic features are linearly decodable from upper layers of transformer language models [5].

4.5 Ablation C: Representation Persistence

If position information degrades before reaching the final token, the model's failure could be attributed to information loss. We tested this by comparing probe accuracy at the step token (where the model just processed the last move) versus the final token (where the model must generate its answer). The results are the opposite of what information-loss would predict. At $N \leq 2$, step-token and final-token probes perform comparably (within 10 percentage points). At $N \geq 3$, final-token probes *outperform* step-token probes by a substantial margin: for GPT-2 Medium at

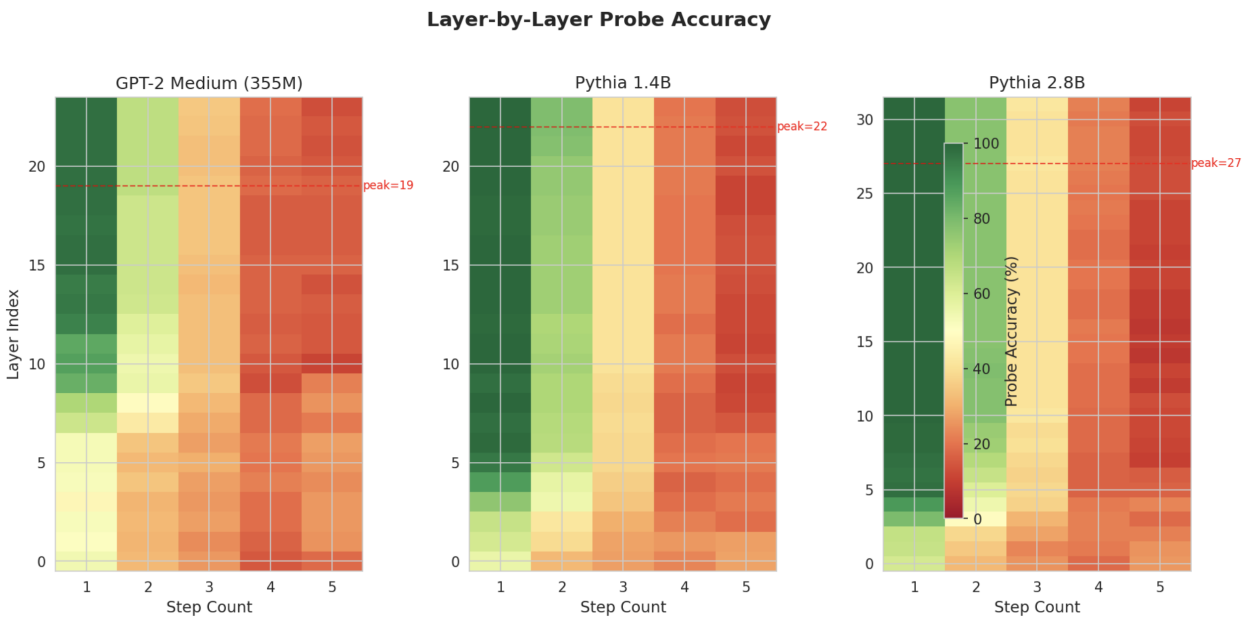

Figure 2: Layer-by-layer probe accuracy (step-token, last step) across all layers and step counts. Dashed red lines mark the peak probing layer. Spatial position information concentrates in upper layers (79–92% depth) and is most strongly encoded at N=1. The pattern is consistent across model scales.

N=5, final-token accuracy is 39.0% versus step-token accuracy of 17.0% ($\Delta_{\text{persist}} = +22.0$ percentage points). Pythia 1.4B shows a similar pattern ($\Delta_{\text{persist}} = +18.0$ at N=5), as does Pythia 2.8B ($\Delta_{\text{persist}} = +18.0$ at N=5). Figure 3 visualizes this crossover: at low step counts, step-token probes are comparable or slightly better; at high step counts, final-token probes dominate.

This result indicates that the model does not simply lose position information as it processes subsequent tokens. Instead, position representations *consolidate* at the final token position. The model aggregates spatial information from the sequence into a representation at the answer position that is more linearly decodable than at any individual step token. Yet despite this consolidation, behavioral accuracy remains near zero. The bottleneck is not information availability but information utilization: the model cannot convert a linearly accessible position representation into the correct output tokens via its unembedding layer.

4.6 Ablation D: Intermediate Position Tracking

For N=5 scenarios, we trained probes at each intermediate step to classify three targets: the *current* position (after that step), the *starting* position (constant across steps), and the *final* position (the endpoint). Figure 4 and Table 3 report the results. Current-position probe accuracy degrades from 96–100% at step 1 to 13–17% at step 5, confirming that the model's spatial tracking deteriorates with sequence length. Starting-position accuracy degrades more slowly, from 100% at step 1 to 36% (GPT-2 Medium), 73% (Pythia 1.4B), and 53% (Pythia 2.8B) at step 5, indicating that the models retain a strong trace of the initial state throughout the sequence. Final-position accuracy remains near chance (8–17%) at all intermediate steps, which is expected: the model has not yet seen all the moves needed to determine the final position.

The strong persistence of the starting position relative to the current position suggests that the models anchor to the initial state rather than maintaining a running spatial update. This asymmetry

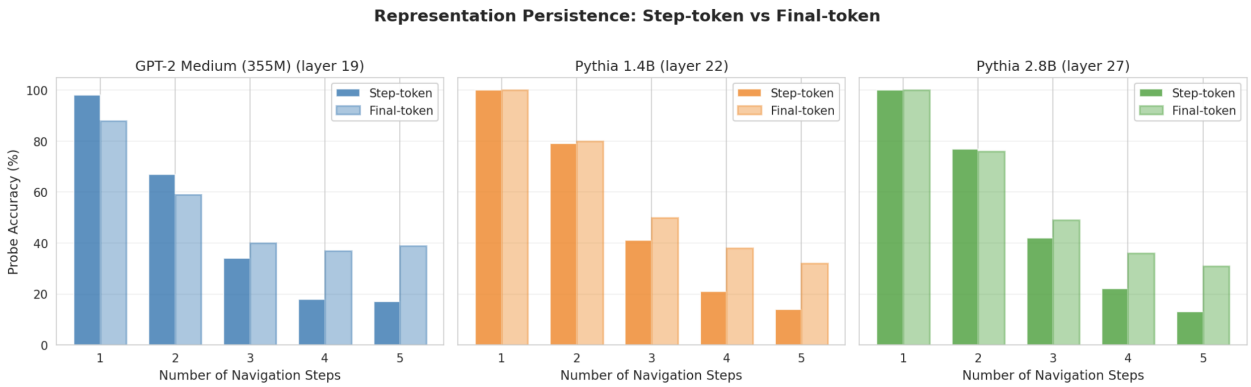

Figure 3: Representation persistence: step-token versus final-token probe accuracy at the peak layer. At $N \geq 3$, final-token probes outperform step-token probes, indicating that position information consolidates at the answer position rather than degrading. The model possesses the information at the point of output generation but cannot produce the correct answer.

Table 3: Intermediate position tracking for N=5 scenarios at the peak probing layer. Probe accuracy (%) for three targets at each step. The starting position persists strongly while the current position degrades rapidly. Final position remains near chance until all moves are observed.

Model	**Target**	Step 1	Step 2	Step 3	Step 4	Step 5
GPT-2 Med.	Current	96.0	73.0	29.0	14.0	17.0
	Start	100.0	99.0	96.0	66.0	36.0
	Final	14.0	14.0	13.0	11.0	17.0
Pythia 1.4B	Current	100.0	83.0	43.0	21.0	14.0
	Start	100.0	100.0	100.0	90.0	73.0
	Final	16.0	9.0	20.0	8.0	14.0
Pythia 2.8B	Current	100.0	83.0	42.0	17.0	13.0
	Start	100.0	99.0	99.0	90.0	53.0
	Final	17.0	10.0	17.0	12.0	13.0

is most pronounced in Pythia 1.4B, where starting-position accuracy remains at 100.0% through step 3 while current-position accuracy has already dropped to 43.0%. The pattern is consistent with a model that encodes the starting position as part of the prompt representation but struggles to compose sequential spatial transformations on top of it.

4.7 Ablation E: Model Scale Comparison

The representation-reasoning gap does not close with model scale across the 355M–2.8B range. At N=1, all three models show near-perfect probe accuracy (98–100%) and near-zero behavioral accuracy (0–5%). At N=3, step-token probe accuracy is 34.0%, 41.0%, and 42.0% for GPT-2 Medium, Pythia 1.4B, and Pythia 2.8B respectively, while behavioral accuracy is 1.2%, 0.0%, and 5.2%. The Pythia models exhibit marginally higher probe accuracy than GPT-2 Medium (likely reflecting the larger residual stream), but behavioral accuracy does not improve. The intermediate tracking results reinforce this: starting-position persistence at step 5 is 36.0%, 73.0%, and 53.0% for the three models; larger models retain the origin more strongly, but current-position tracking at

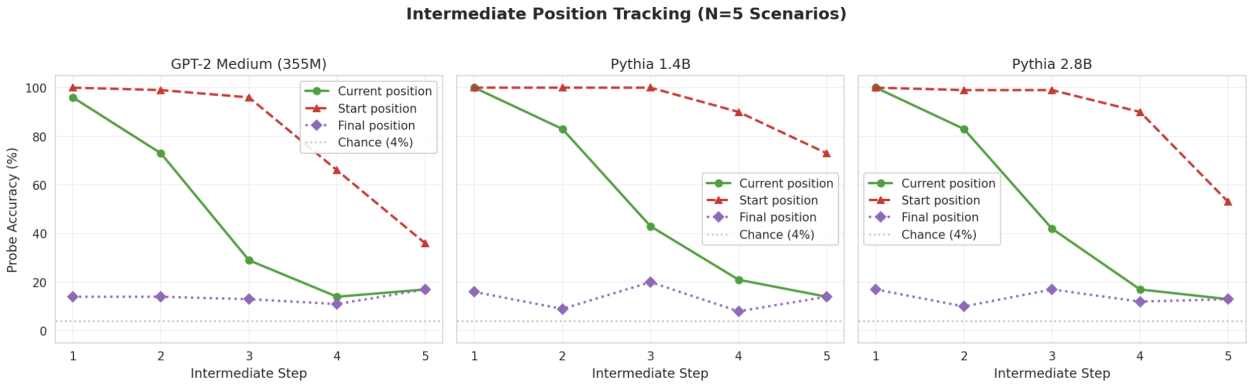

Figure 4: Intermediate position tracking for N=5 scenarios at the peak layer. Current position degrades from 96–100% to 13–17% across five steps. Starting position persists more strongly (100% $\rightarrow$ 36–73%), suggesting anchoring to the initial state. Final position remains near chance throughout, as expected.

step 5 converges to 13–17% regardless of scale. The 8$\times$ increase from 355M to 2.8B parameters does not bridge the representation-reasoning gap, suggesting that the failure mode is architectural or algorithmic rather than a matter of model capacity at this scale.

5 Discussion

The central finding of this work is a large, persistent dissociation between representation formation and compositional reasoning in language models performing spatial navigation. At the representational level, models encode the agent's position with near-perfect linear decodability after a single move. At the behavioral level, they cannot produce the correct position as output, even in the simplest N=1 case. This dissociation is not a failure of information transmission: the persistence analysis demonstrates that spatial information *consolidates* at the final token, becoming more linearly accessible at the point where the model must generate its answer. The bottleneck therefore lies in the mapping from internal representation to output tokens, specifically the unembedding and output projection stages, rather than in the formation or maintenance of the representation itself.

This finding refines the conclusions of prior work on world models in language models. Gurnee and Tegmark [5] showed that LLaMA-2 encodes geographic coordinates; Li et al. [7] showed that Othello-GPT encodes board state. Our results suggest a complementary interpretation: representation formation may be a general byproduct of next-token prediction that does not require, and does not enable, deliberate reasoning over the represented structure. The compositionality gap documented by Press et al. [10] and the transformer limitations identified by Dziri et al. [3] may stem not from a failure to represent compositional structure, but from a failure to read out and act on representations that are already present.

The non-monotonic behavioral accuracy pattern (higher accuracy at N=2 and N=4 than at odd step counts) is unexpected and warrants further investigation. One possible explanation is that even-length move sequences are more likely to produce positions near the starting point (pairs of moves can cancel), and models default to predicting positions near the stated start. This would constitute a form of surface-level pattern matching rather than spatial computation, consistent with the general finding that these models are not performing genuine spatial reasoning.

Limitations. Several caveats apply to these results. First, linear probe accuracy measures linear *accessibility*, not causal *use*. A high-accuracy probe demonstrates that position information is present in the activation geometry, but does not prove the model uses this information in its forward computation. Causal interventions such as activation patching would provide stronger evidence but are beyond the scope of this study. Second, the 5×5 grid-world is a synthetic and simplified proxy for spatial reasoning. Results may not generalize to naturalistic spatial descriptions or more complex environments. Third, we tested only three model sizes (355M, 1.4B, 2.8B); the scaling trend is suggestive but not definitive, and larger models (7B+) may close the gap. Gated access prevented testing larger models such as LLaMA-3-8B or Gemma-2-2B. Fourth, the RidgeClassifier used for probing provides a closed-form linear solution that may be slightly less expressive than multinomial logistic regression, though the 98–100% accuracy at N=1 suggests this is not a practical limitation. Fifth, probe accuracy is an upper bound on what is linearly extractable from the activations; the model's own computational access to this information via its weight matrices may be weaker. Sixth, the grid-world task conflates spatial representation with sequential instruction following, and some observed failures may reflect difficulty tracking a sequence of instructions rather than spatial reasoning per se. Seventh, the non-monotonic odd-even pattern in behavioral accuracy is unexplained and may indicate a confound in how models pattern-match on prompt structure. Eighth, the final position distribution is non-uniform due to boundary effects in the rejection-sampling procedure (corner cells are overrepresented as final positions relative to center cells), which could inflate probe accuracy on frequently occurring positions; however, the 98–100% accuracy at N=1 across all positions suggests this is not a dominant factor. Finally, the small test set ($n = 100$) means individual probe accuracy estimates carry approximately ±5–8 percentage points of uncertainty, though this does not affect the qualitative conclusions given the magnitude of the gap.

Implications. If the representation-reasoning gap is a general phenomenon, it motivates architectural modifications that explicitly bridge internal representations and output generation. Chain-of-thought prompting, scratchpads, and external memory systems all provide mechanisms for models to externalize intermediate reasoning steps, potentially bypassing the bottleneck in direct representation-to-output mapping. The finding that representations consolidate at the final token is particularly suggestive: the information is available at exactly the right position, in a linearly decodable form, but the model's output projection cannot make use of it. This points to the unembedding layer, or the lack of iterative computation between representation and output, as a key locus of failure.

6 Conclusion

We have demonstrated a representation-reasoning gap in language models performing spatial navigation: linear probes recover the agent's grid position with 98–100% accuracy from residual stream activations, while the same models produce the correct answer 0–5% of the time. This gap is maximal at one step (where the task is simplest), persists across step counts, and does not close with model scale from 355M to 2.8B parameters. Position representations consolidate at the final token position, ruling out information loss as the primary failure mode. These findings establish that representation formation is cheap, a byproduct of language modeling that emerges even in small models, while reasoning over those representations is the hard problem. Future work should apply causal interventions (activation patching) to test whether the identified representations are causally involved in the model's computation, extend the analysis to larger gated models, and investigate

whether the representation-reasoning gap generalizes to naturalistic spatial tasks.

References

[1] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

[2] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2023.

[3] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D Hwang, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, 2023.

[4] Joshua Engels, Isaac Liao, Eric J Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are linear. In *International Conference on Machine Learning*, 2024.

[5] Wes Gurnee and Max Tegmark. Language models represent space and time. In *International Conference on Learning Representations*, 2024.

[6] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

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

[8] Neel Nanda, Andrew Lee, and Martin Berber. Actually, othello-GPT has a linear emergent world representation. *arXiv preprint arXiv:2310.07582*, 2023.

[9] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, 2024.

[10] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

[11] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.