

# Language Models Struggle to Use Representations Learned In-Context

Michael A. Lepori<sup>†\*</sup>

Tal Linzen<sup>¶§</sup>

Ann Yuan<sup>‡</sup>

Katja Filippova<sup>‡</sup>

<sup>†</sup> Brown University

<sup>¶</sup> Google Research

<sup>§</sup> New York University

<sup>‡</sup> Google DeepMind

## Abstract

Though language models (LMs) have enabled great success across a wide variety of tasks, they still appear to fall short of one of the loftier goals of artificial intelligence research: creating an artificial system that can adapt its behavior to radically new contexts upon deployment (Shi et al., 2024). One important step towards this goal is to create systems that can induce rich representations of data that are seen in-context, and then flexibly deploy these representations to accomplish goals (Lampinen et al., 2024). Recently, Park et al. (2025a) demonstrated that current LMs are indeed capable of inducing such representation from context (i.e., in-context representation learning). The present study investigates whether LMs can use these representations to complete simple downstream tasks.

We first assess whether open-weights LMs can use in-context representations for next-token prediction, and then probe models using a novel task, **adaptive world modeling**. In both tasks, we find evidence that open-weights LMs struggle to deploy representations of novel semantics that are defined in-context, *even if they encode these semantics in their latent representations*. Furthermore, we assess closed-source, state-of-the-art reasoning models on the adaptive world modeling task, and demonstrate that even the most performant LMs cannot reliably leverage novel patterns presented in-context. Overall, this work seeks to inspire novel methods for encouraging models to not only encode information presented in-context, but to do so in a manner that supports flexible deployment of this information.

## 1 Introduction

The advent of transformer-based language models (LMs) has radically changed how we think about the capabilities of deep neural networks. Tasks

that once appeared out of reach, such as competition mathematics (Luong et al., 2025), multilingual question answering (Comanici et al., 2025), and expert-level scientific reasoning (Rein et al., 2024), have all seen substantial progress as a direct result of LM advancements. These advanced capabilities have spurred an ongoing investigation into whether transformers trained on next-token prediction encode an internal model of the causal process that generated their training data (i.e., a “world model”), an internal model that could support such inferences (Li et al., 2021; Toshniwal et al., 2022; Li et al., 2023; Vafa et al., 2024, 2025). Artificial agents that are endowed with such internal world models may ultimately prove more self-consistent (Wong et al., 2025), generate fewer hallucinations (Wong et al., 2023), and generalize better beyond their training distribution (Wu et al., 2024; McCoy et al., 2024; Gupta et al., 2025).

In order to ensure that models can operate in a dynamic world without retraining, we would prefer artificial agents that are capable of adapting to different environments upon deployment. LMs have demonstrated a form of this ability, exemplified by the emergence of in-context learning (ICL; Brown et al. 2020). ICL enables pretrained LMs to temporarily adopt new behaviors based on examples or descriptions provided in context, without updating the model’s parameters (Davidson et al., 2025). A fully realized adaptable agent would be able to learn to represent rich, structured representations that are defined entirely in-context *and* flexibly deploy<sup>1</sup> those representations to solve downstream tasks (Lampinen et al., 2024). In other words, such

<sup>1</sup>We consider a representation to be **deployable** for a particular context if it plays a causal role in responding to that context, otherwise (e.g., if the representation is decodable, but not used; Dechamps and Zador (2000)) we consider the representation to be **inert** with respect to that context. A representation is **flexibly deployable** if it is deployable for contexts distinct from the one that gave rise to it (e.g., downstream tasks).

\*Corresponding author: michael\_lepori@brown.edu; Work done while at Google DeepMind.

an agent would be able to construct an *in-context world model*.

A recent study explicitly investigates whether LMs can learn the kinds of structured internal representations required by an in-context world model. Park et al. (2025a) present LMs with a random walk over a state space with a latent topology, where each state is denoted by an arbitrary token (See Figures 1 and 2 for examples of these topologies). They demonstrate that LM token representations come to reflect the topology of that state space, moving the representations away from their default, uncontextualized semantics. Thus, LMs can in-context learn representations that encode completely novel semantics, a finding which has been replicated (Lubana et al., 2025) and used as inspiration for practical LM interventions (Yona et al., 2025). Here, we ask: are these in-context representations flexibly deployable when the model is faced with novel tasks, or are they inert in these conditions (Vafa et al., 2024; Li et al., 2025b)? See Figure 1 for an illustration.

We investigate this question using prompts which crucially have two components: one which defines the in-context semantics (colored words in Figure 2) and a separate component which requires the model to refer to these semantics to complete a task. We focus on two simple, synthetic tasks: (i) a next-word prediction task (Figure 2 Right), and (ii) a novel task, **Adaptive World Modeling** (Figure 2 Bottom Left). We find that in-context representations present in the hidden states of an LM are largely inert.

However, LMs may represent structures defined in context without encoding them in the geometry of their hidden states. We investigate one particularly salient alternative, exemplified by state-of-the-art closed-source LMs. These so-called “reasoning models” are trained to externalize their intermediate reasoning steps into long chains of thought (CoTs; Wei et al. 2022; Nye et al. 2021; Jaech et al. 2024). Crucially, they appear to process information in a somewhat different manner than their base LM counterparts (Bogdan et al., 2025), with CoT techniques enabling them to dynamically reason about a breadth of possible approaches to a given task (DeepSeek-AI et al., 2025; Yao et al., 2023), and even to behave as if they had constructed a model of the scenarios underlying a task (Coda-Forno et al., 2024). We investigate the possibility that these externalized reasoning chains can compensate for otherwise-inert hidden representations,

enabling the creation of in-context world models. We find evidence that frontier reasoning models encode novel, in-context semantics in a manner that sometimes facilitates their use when solving downstream tasks, but they do so imperfectly, leaving substantial room for improvement.

In summary, our contributions are as follows:

1. We introduce a new task, **adaptive world modeling**, that is designed to probe whether LMs can flexibly deploy novel, in-context semantics to solve simple tasks (Section 4).
2. We find that open-weights models struggle to use in-context semantics to solve either next-token prediction tasks (Section 3) or adaptive world modeling tasks (Section 4), *even when their latent representations encode these semantics*.
3. We find that frontier reasoning models achieve limited success on adaptive world modeling tasks, suggesting that their use of long externalized reasoning chains may partially enable the use of in-context semantics (Section 5).

## 2 Preliminaries: In-Context Representation Learning

**Graph Tracing Task** We study LMs’ in-context representation learning capabilities using the graph tracing task introduced by Park et al. (2025a). The task is to predict the next word in a sequence of common, semantically unrelated one-token words. The sequence is generated by a random walk over a latent state space, with each state denoted by a unique word (See Figure 2). In our experiments, these state spaces are organized into one of four topologies, defined by their dimensionality (2D grids or 1D lines) and total number of states (16 or 25). Specifically, we study 4-by-4 grids, 5-by-5 grids, 16-state lines, or 25-state lines, with many random assignments of words to states. As in prior work, we present a random walk over 2D topologies, and present a sequence of randomly selected pairs of adjacent states for 1D topologies. We measure how the representations of the words change as a function of context length. Using the metrics described below, we replicate Park et al. (2025a)’s main result: latent representations come to recapitulate the geometry of the underlying state space after sufficient context.

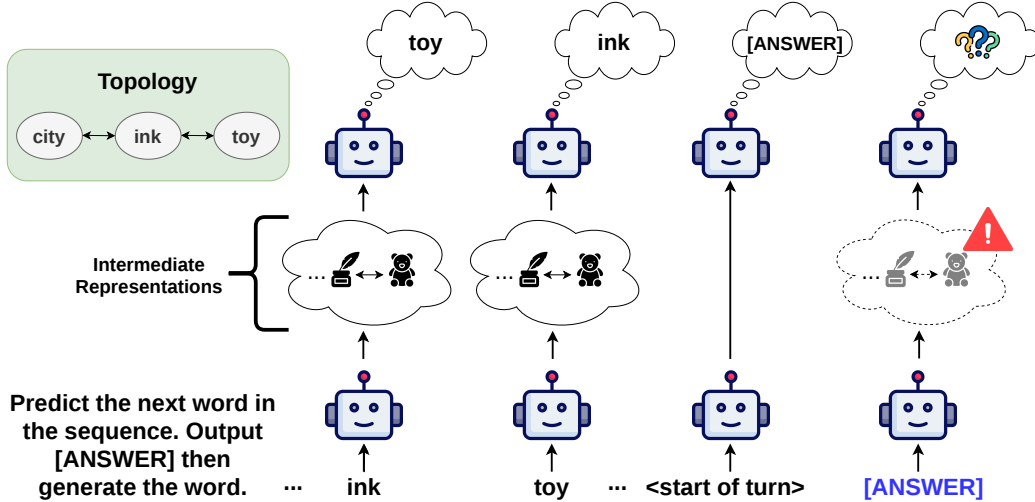


Figure 1: An illustration of a structured representation that is learned in context, but is not flexibly deployable. In the next-token prediction task, we present models with sequences of arbitrary words generated by a random walk over a topology (in green), and ask the model to generate an [ANSWER] token, followed by the next token. The prompt tokens are represented in black, and the model-generated tokens are in blue. Given such a sequence, a language model may encode a representation of that topology within some of its hidden states after some number of layers. However, the model may be unable to properly deploy this representation later on when it is necessary for generating a response. In this work, we find that representations that are learned in context are oftentimes not flexibly deployable.

**Metrics** Given a set of LM hidden representations corresponding to every token in the state space, we wish to characterize the extent to which these hidden states reflect the topology of the state space. Following prior work, we preprocess the representations by taking the mean of the hidden representations of all instances of each word in the state space at a given layer  $l$  over a sliding window of 50 tokens in the random walk (or 50 adjacencies for 1D topologies). This gives us an average representation for each word at that layer,  $H^l(\mathcal{T})$ . We characterize in-context representation learning using two metrics:

**Dirichlet Energy (DE):** This metric measures the distance between representations whose corresponding words are adjacent in the underlying state space. The core idea is that in-context representation learning should push adjacent words to be similar to one another. DE is defined as

$$E_G(H^l(\mathcal{T})) = \sum_{i,j} A_{i,j} \|h_i^l - h_j^l\|^2 \quad (1)$$

where  $E_G$  is the DE given state space topology  $G$ ,  $A_{i,j}$  is an indicator variable that denotes whether tokens  $i$  and  $j$  are adjacent in the state space, and  $h_i^l$  refers to a particular hidden state of token  $i$  at layer

$l$ . We normalize this metric by the total energy of the state space:  $\sum_{i,j} \|h_i^l - h_j^l\|^2$ .

**Distance Correlation (DC):** This metric measures the correlation between distances in representation space and distances in the underlying state space. States that are far apart in the state space should correspond to representations that are far apart. Formally, DC is defined as

$$D_C(H^l(\mathcal{T})) = \text{Corr}(D_R^l, D_G) \quad (2)$$

where

$$D_{R_{i,j}}^l = \|h_i^l - h_j^l\|_2 \quad \text{and} \quad D_{G_{i,j}} = \|G_i - G_j\|_1 \quad (3)$$

$G_i$  refers to the coordinates of the token  $i$  in the underlying state space. In other words, DC is the correlation between the Euclidean distance of the representations of particular tokens and the Manhattan distance between the states that these tokens correspond to.

**Replication** We study four capable instruction-tuned open-weights models, gemma-3-{4, 12, 27}b-it (Team et al., 2025) and OLMo-2-13b (OLMo et al., 2024). In general, we find that models can successfully learn representations in-context — as context length increases, representations increasingly conform to the underlying state

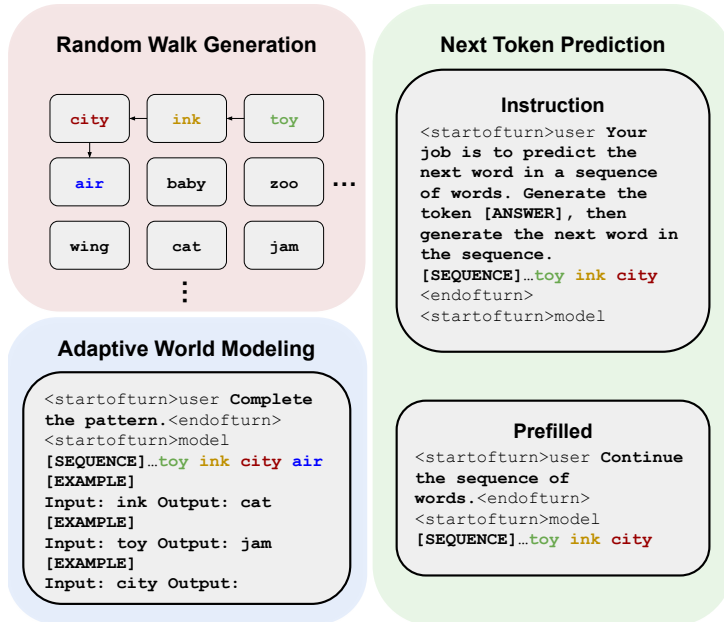


Figure 2: **(Top Left)** Example of an N-by-N state space topology used to generate a random walk. **(Right)** Examples of next token prediction prompts in the **Instruction** or **Prefilled** condition. Prompt formatting tokens are not bolded for readability. In the Instruction condition, models need to deploy in-context representations that are formed during the random walk (colored tokens) after an interval of several tokens in order to predict a valid next token. In the Prefilled Condition, in-context representations are deployed immediately. **(Bottom Left)** Example of an adaptive world modeling prompt, which consists of a random walk (colored tokens), followed by few-shot examples defining a rule that maps states at one position to states at another. Here the rule maps states  $s_{i,j}$  to  $s_{i+2,j}$  (a “two-step rule”). Here, *city* would get mapped to *wing*.

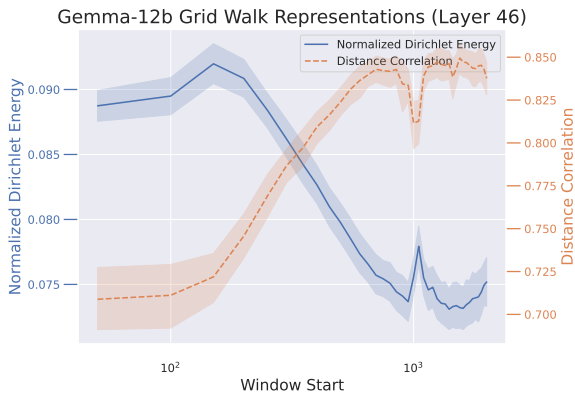


Figure 3: Example of in-context representation learning over a 5-by-5 grid topology. **Dirichlet Energy** decreasing and **Distance Correlation** increasing indicates better in-context representation learning.

space topology. See Figure 3 for results taken over 50 different token assignments on a 5-by-5 grid for an arbitrary late layer in gemma-3-12b-1t. We note some variation in the dynamics of in-context representation learning for some model-topology combinations, including some combinations that exhibit in-context representation learning early and then degrade (See Appendix A).

### 3 Experiment 1: In-Context Representation Learning Does Not Imply Robust Next-Token Prediction

To assess whether in-context representations are flexibly deployable, we take inspiration from prior work that identified a gap in next-word prediction performance when LMs are forced to delay their predictions (Hu and Frank, 2024). While this work studied LM behavior on naturalistic sentences, we propose to extend their results into our synthetic graph tracing setting where we can quantify whether the representations encode the semantics required to perform next-word prediction. This allows us to disentangle failures of encoding semantics from failures of deploying these semantics.

Specifically, we vary whether the random walk is presented in a user message (Instruction Condition), or whether it is presented in a prefilled model response (Prefilled Condition; See Figure 2 Right). In the Instruction Condition, the model is instructed to predict the next word in its response, which requires the model to delay using the in-context learned representations (due to the presence of intermediate special tokens). In contrast, the Pre-

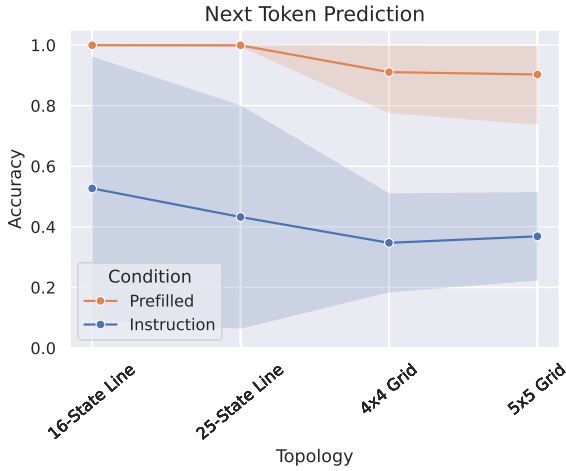


Figure 4: Next-token prediction results for all open-weights models over all topologies. Models struggle when the random walk is present in the user prompt, rather than in a prefilled model response. This indicates that the representations learned in-context during the random walk/random adjacencies are not easily deployed when they need to be used at a later time.

filled Condition requires that the model uses the in-context representations immediately to perform next-word prediction. A response is considered “correct” if it is in the set of valid next tokens, given the preceding sequence and underlying topology.

**Results** For each model and state space topology used in Section 2, we identify the context length the maximizes the average distance correlation achieved in the last and third-to-last layers (See Appendix B for the specific values). We run the next-word prediction analysis with 1000 different word assignments across all models and all state space topologies. Our results point to a disparity between conditions: models reliably predict a valid next-word in the random walk when the random walk is prefilled (replicating prior work (Park et al., 2025b; Lubana et al., 2025)), but struggle when the random walk is presented in the user prompt (See Figure 4).

**Analysis** One potential explanation for the performance disparity is that in-context representation learning simply yields worse representations in the Instruction Condition than the Prefilled Condition. However, we demonstrate that this is not the case in Figure 5. We compute the ratio of both in-context representation learning metrics over all models and topologies across all context windows (comprising 50 states in a random walk or 50 sampled adjacencies, depending on the dimensionality of the



Figure 5: Ratio of in-context representation learning metrics between prompt format conditions over context. There does not appear to be a substantial difference between conditions (i.e., the ratio of metrics is near 1). If there is a small difference, it appears to be in-favor of instruction prompt representations, rather than prefilled prompt representations.

underlying topology). We find that in-context representation learning occurs in both conditions approximately equally. In fact, the small distinction that exists suggests higher fidelity representations in the Instruction Condition, thus ruling out the possibility that the Instruction Condition results in worse representations than the Prefilled Condition.

One might also worry that the context lengths we selected are insufficient for accurate next-token prediction. Perhaps in-context representations are *initially* inert, but then become flexibly deployable after additional context. Indeed Park et al. (2025a) note that accurate next-token prediction sometimes requires a longer context length than in-context representation learning. We investigate this possibility in Appendix C, and find that models still struggle in the Instruction Condition with extended context lengths. Taken together, these results suggest that representations that are learned in-context are largely inert.

## 4 Experiment 2: In-Context Representation Learning Does Not Imply Adaptive World Modeling

In Section 3, we identified a subtle divergence between in-context representation learning and next-token prediction performance, suggesting that in-context representations are not flexibly deployable. Here, we define a task, **Adaptive World Modeling (AWM)** to further probe this.

**Task** AWM prompts consists of two components. The first is the graph tracing task (defined in Section 2). Models must infer the latent state space topology from a random walk over that topology. The second component is a few-shot learning task, where elements at one position in the topology get mapped to elements at another position according to a simple rule (e.g., state  $s_{i,j} \rightarrow s_{i+2,j}$ ; See Figure 2). Because completely arbitrary words are used to denote each state, LMs *must* deploy the structure that they have encoded in-context to solve the task.

We investigate the same set of topologies presented in Section 2, and investigate several different rules that govern the few-shot learning task. For all topologies, we investigate a “1-step down” rule (where the input-output transitions are attested in the random walk) and “2-step down” rule (where they are not; see Figure 2). For the 5-by-5 grid, we additionally investigate a “3-step rule”, where states at position  $(i, j)$  get mapped to states at position  $(i + 2, j + 1)$ .

In most settings, we include 10 examples in the context and query the model with a held-out example. Because the 4-by-4 grid topology only contains 8 well-defined “2-step down” examples, we include 6 examples for that prompt. We present the prompts as prefilled model responses<sup>2</sup> and use the context lengths used in Section 3. We evaluate on 1000 token assignments for each condition.

**Results** From Figure 6, we see that all open-weight LMs that we tested struggle with the AWM task. LMs may seem to perform somewhat better at the one-step rules, but these rules allow for a trivial baseline strategy of randomly guessing a transition observed in the random walk, rather than truly inferring the underlying rule (see Appendix D for details). However, we do see that some topology-rule combinations permit nontrivial performance on the AWM task. In particular, models achieve approximately 40% accuracy on the 4-by-4 grid topology for one-step and two-step rules.

**Analysis** An LM that has properly formed in-context representations corresponding to the underlying state space might struggle to solve an AWM task for one of two reasons: (i) the LM cannot properly induce the rule from the few-shot examples,

<sup>2</sup>We also tested `gemma-3-27b-it` on a version of the task where the instructions, random walk, and few-shot examples are presented in the user message, and achieve approximately the same poor performance.

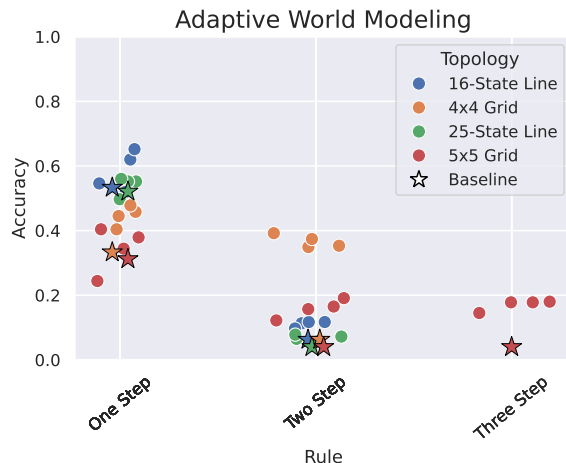


Figure 6: Adaptive World Modeling results. Across all task configurations, various open-weights LMs (each represented by a separate dot) struggle, despite encoding the underlying grid topology in their latent representations. Note: A naive strategy of randomly selecting an attested adjacency would achieve approximately 50% (for one-dimensional) and 30% (for two-dimensional) accuracy for the “one-step” rules.

even with perfect information about the topology, or (ii) the in-context learned representations are largely inert in the context of an AWM task, i.e., the LM cannot reliably use the topology information encoded in them when confronted with the few-shot examples. We investigate both of these alternatives.

First, we investigate the model’s few-shot learning ability. We provide an explicit description of the underlying state space for the 4-by-4 or 5-by-5 grid topologies in the prompt and then provide few-shot examples of the rule before querying the model with a held-out state. Specifically, we include statements of the form Coordinates:  $\emptyset$   $\emptyset$  Item: city to describe the grid topology. In Figure 7 (Top), we see that LMs achieve substantially higher accuracy when given the explicit description than when they need to infer the topology in-context, with some models achieving  $> 75\%$  accuracy on AWM tasks defined on the 4-by-4 grid with one-step and two-step rules. Thus, LMs’ struggles with AWM cannot be fully attributed to their ability to learn the rule from examples.

Next, we investigate whether the representations learned in-context are flexibly deployed during the few-shot learning portion of the AWM task. In particular, we investigate whether the hidden representations of words present in the few-shot examples also encode the state space topology. If representa-

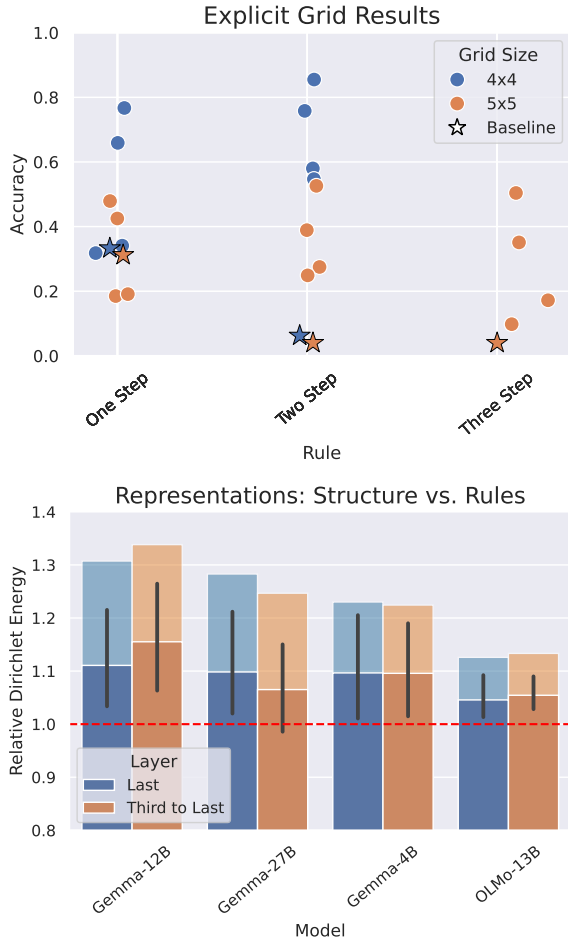


Figure 7: (Top) Open-weights LM results on few-shot learning when the state space topology is explicitly presented in the prompt. LMs achieve variable performance, but it is typically much higher than in the AWM setting. (Bottom) Relative Dirichlet Energy between tokens in the random walk and in the few-shot examples (and between random walk tokens and uncontextualized tokens; light bars) over two layers. In-context representations encode the underlying topology with lower fidelity in the few-shot examples.

tions do not reflect the state space topology when words are presented in the few-shot learning examples, then this provides evidence that in-context representations are not being flexibly deployed to learn the rule. We measure the ratio of the Dirichlet Energy (DE) of representations drawn from the random walk to the DE of the same tokens when they are present in the few-shot examples. If this ratio is greater than 1, then the representations present in the examples encode the underlying topology with lower fidelity than the representations in the random walk (see Appendix E for details). For comparison, we also compute the ratio of the DE of uncontextualized tokens to the DE of random

walk tokens. In Figure 7 (Bottom), we find that representations in the few-shot learning examples are degraded relative to their counterparts in the random walk. This suggests that the in-context learned representations are not being flexibly deployed to solve the adaptive world modeling task.

One might be concerned that LMs simply do not infer that the few-shot examples are at all related to the random walk. We test a meta-learning variation of the task where examples of successful completions are prepended to the prompt, but find that models still perform poorly (See Appendix F). Similarly to Section 3, we additionally investigate longer contexts in Appendix G, and find that this also does not improve performance. Taken together, these results further support the claim that representations learned in-context are largely inert (i.e., unavailable for solving downstream tasks).

### 5 Experiment 3: Investigating Reasoning Models

Finally, we investigate whether our analyses generalize to frontier models: Gemini-Flash-2.5, Gemini-Pro-2.5 (Comanici et al., 2025), GPT-5-mini, and GPT-5 (Team, 2025). These models are so-called “reasoning models”, which generate extended, verbalized reasoning chains before providing an answer to a user query (Nye et al., 2021). This form of computation represents the state-of-the-art for modern LMs and has resulted in substantial improvements in tasks that require advanced reasoning (Comanici et al., 2025; Team, 2025; DeepSeek-AI et al., 2025).

However, the representations and mechanisms supporting reasoning models remain even more opaque than the inner workings of standard LMs (Bogdan et al., 2025; Macar et al., 2025), with some arguing that reasoning models implement fundamentally new algorithms when using extended reasoning chains (Li et al., 2025c). Can these advanced reasoning models offer a means of encoding in-context semantics and flexibly deploying them to solve downstream tasks? In this section, we investigate this question by testing frontier LMs on the AWM task.

**Results** Similarly to our open-weight model analyses in Appendices C and G, we provide models with 1500-token random walks or 1500 pairs of adjacent states for each topology, and use the same rules as described in Section 4. For Gemini models, we let the model generate up to 5000 tokens

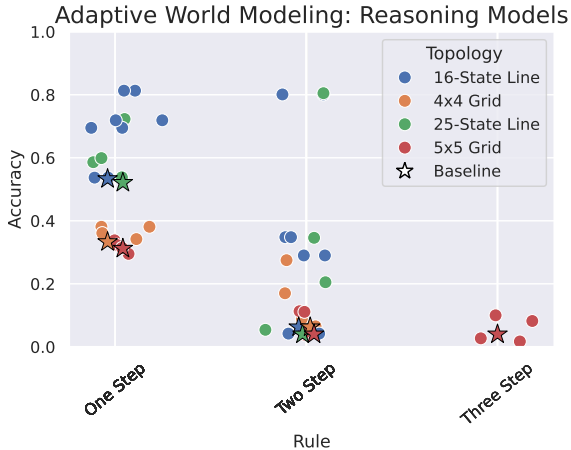


Figure 8: Frontier LM results on AWM. Frontier LMs achieve higher performance than open-weights models, especially for one-dimensional (line) topologies. However, performance drastically degrades when they are confronted with two-dimensional (grid) topologies.

in its chain-of-thought before responding. We impose this “thinking budget” to prevent timeouts caused by extremely long thinking generations. For GPT-5 models, we enable the model to flexibly determine its thinking budget. We do not view this discrepancy as an issue, as our primary question is whether *any* LMs can perform well on the adaptive world modeling task. Furthermore, initial experimentation revealed that imposing a 5000-token thinking budget did not substantially impact the performance of Gemini models.

Our results are presented in Figure 8. We see that even these frontier reasoning LMs do not completely solve the AWM task, though many perform markedly better than their open-weights counterparts. In particular, some models achieve nontrivial accuracy on 1D topologies for one-step and two-step rules. However, accuracy entirely collapses when models are presented with grid topologies.

**Analyses** Similarly to Section 4, we investigate two potential sources for LM errors on the AWM task: 1) Not being able to induce the rule from the few-shot learning examples (even with perfect knowledge of the underlying state space topology) and 2) Not being able to produce a representation of the underlying state space topology that is able to be deployed for downstream tasks. We investigate Gemini-2.5-{Pro, Flash}.

First, we investigate whether frontier reasoning LMs can solve the few-shot learning task when they are presented with an explicit representation of the underlying state space. This analysis proceeds

exactly as in Section 4. From Figure 9, we see that both models achieve ceiling performance in this setting. Thus, errors are *not* caused by a failure to learn the rule from the examples.

Next, we investigate whether these models can induce a flexible representation of the underlying state space topology. Because we do not have access to the hidden representations of these frontier models, we investigate this question by directly asking the model to describe the structure of the state transition matrix underlying a random walk. We provide models with 1000 random walks over 4-by-4 and 5-by-5 grids. Succeeding at this requires the model to (i) have inferred the underlying topology in the first place and (ii) be able to deploy these encoded in-context semantics to generate a natural language description of them. To assess whether the model successfully describes the state space, we employ a permissive autorating strategy. Specifically, we provide each LM response in a new prompt to Gemini-2.5-Pro, and ask the model to respond “yes” if the response contains a reference to a 4-by-4 (or 5-by-5) grid, and otherwise respond “no”. Note that the original LM description does not need to actually infer the correct word-state mapping to count as correct. Nonetheless, we find that both Pro and Flash models achieve good, but not perfect performance (See Figure 9). By construction, the LMs’ rate of achieving a perfect understanding of the underlying state space is even lower. Thus, it appears that errors are driven primarily by the LMs’ encoding and deployment of a flexible representation of the state space topology.

In Appendix H, we present frontier model results from the next-token prediction task. We find that most of these models perform reasonably well. In Appendix I we investigate the role of reasoning chains on task performance for both next-token prediction and adaptive world modeling by ablating the reasoning chains in Gemini models. We find mixed results, with performance on both tasks sometimes substantially *increasing* for Flash with reasoning chains ablated, and performance on both tasks systematically dropping for Pro. Finally, in Appendix J we find that additional guidance on the underlying state space topology improves model performance, providing further evidence that representations of the underlying topology enable reasoning models to succeed at AWM.

Taken together, these results suggest that externalized reasoning chains may provide an alternative pathway for models to encode and deploy repre-

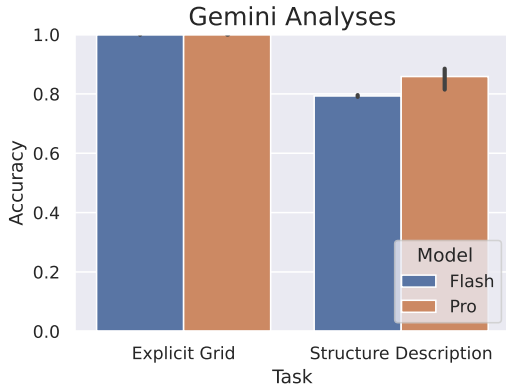


Figure 9: (Left) When presenting either Gemini model with a prompt that explicitly describes the underlying state space topology, the LM achieves ceiling accuracy on a few shot learning task. (Right) When prompting Gemini models to describe the structure of the latent state space that generated a random walk, models achieve fair, but not perfect, accuracy (even when assessed using a lenient autorater).

sentations of semantics that are defined in-context. However, the circumstances in which these representations may be deployed is somewhat limited, leaving much room for improvement in the development of adaptable artificial intelligence systems.

## 6 Discussion

**Related Work** The present work is situated at the intersection of two active research directions: (i) characterizing the internal world models implemented by language models and (ii) characterizing the capabilities and limitations of in-context learning. Several works have evaluated world models by probing their internal representations (Li et al., 2023; Gurnee and Tegmark, 2024; Li et al., 2021) or generalization behavior (Vafa et al., 2024, 2025). Many of these works rely on a model’s ability to track state throughout context, a precursor to the in-context representation learning under study in the present work. Finally, several theoretical works have attempted to define a world model (Andreas, 2024; Li et al., 2025a). One common theme is that a world model is canonically *not* just a look-up table, and instead affords other manipulations. In the present context, this implies that an inert set of representations learned in-context does not constitute an in-context world model.

With respect to in-context learning, much research has focused on characterizing ICL’s algorithmic properties (Akyürek et al., 2023; Von Oswald et al., 2023), mechanistic implementation (Olsson

et al., 2022; Todd et al., 2024; Hendel et al., 2023), and the conditions for its emergence (Chan et al., 2022; Park et al., 2025b; Wurgaft et al., 2025). However, the study of in-context representation learning is still in its infancy (Park et al., 2025a; Lubana et al., 2025; Yona et al., 2025).

**Conclusion** In this work, we investigated whether LMs could flexibly deploy representations of in-context defined semantics to solve downstream tasks. Across several tasks and models, our results suggest that LMs cannot reliably do so. Crucially, even when LMs encode these semantics in their latent representations, these representations are often *inert*, or unable to be used for either next-token prediction or adaptive world modeling. Though frontier models seem somewhat more adaptable than their non-reasoning counterparts, the creation of truly adaptable artificial agents requires training regimens and architectures that support flexible deployment of representations learned in-context.

## 7 Limitations

This work has several limitations. Notably, we reveal that in-context semantics encoded in hidden representations may not be flexibly deployable, but do not offer a solution. We speculate that either targeted training regimens or mechanistic interventions that make these semantics available to earlier model layers (similar to those found in Biran et al. (2024) and Lepori et al. (2025)) can partially address this issue.

Furthermore, it is difficult to understand the relationship between the externalized chains-of-thought produced by reasoning models and their hidden token representations. While we provide evidence that models that employ chain-of-thought seem to partially overcome the limitations of in-context learned representations, we do not yet understand the mechanisms by which this occurs. Future work might focus on clarifying this relationship.

## Acknowledgments

The authors would like to thank Andrew Lampinen, Martin Wattenberg, Asma Ghandeharioun, Effie Li, Eghbal Hosseini, Yasaman Bahri, Julian Zimmert, and the members of the Computation and Psycholinguistics Lab at NYU for their helpful feedback. The authors would also like to thank Rachel Goepner, for proofreading the manuscript.

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 2439559. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

- Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*.
- Jacob Andreas. 2024. Language models, world models, and human model-building.
- Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. 2024. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14113–14130.
- Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. 2025. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891.
- J Coda-Forno, M Binz, JX Wang, and E Schulz. 2024. Cogbench: a large language model walks into a psychology lab. In *41st International Conference on Machine Learning (ICML 2024)*, pages 9076–9108.
- Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Guy Davidson, Todd M Gureckis, Brenden M Lake, and Adina Williams. 2025. Do different prompting methods yield a common task representation in language models? *arXiv preprint arXiv:2505.12075*.
- R Christopher Decharms and Anthony Zador. 2000. Neural representation and the cortical code. *Annual review of neuroscience*, 23(1):613–647.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*. *ArXiv*, abs/2501.12948.
- Prakhar Gupta, Henry Conklin, Sarah-Jane Leslie, and Andrew Lee. 2025. Better world models can lead to better post-training performance. *arXiv preprint arXiv:2512.03400*.
- Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*.
- Roe Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Jennifer Hu and Michael Frank. 2024. Auxiliary task demands mask the capabilities of smaller language models. In *First Conference on Language Modeling*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Andrew Kyle Lampinen, Stephanie CY Chan, Aaditya K Singh, and Murray Shanahan. 2024. The broader spectrum of in-context learning. *arXiv preprint arXiv:2412.03782*.
- Michael A Lepori, Michael Curtis Mozer, and Asma Ghandeharioun. 2025. Racing thoughts: Explaining contextualization errors in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3020–3036.
- 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.
- 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. *ICLR*.
- Kenneth Li, Fernanda Viégas, and Martin Wattenberg. 2025a. What does it mean for a neural network to learn a "world model"? *arXiv preprint arXiv:2507.21513*.

- Yuxuan Li, Declan Iain Campbell, Stephanie CY Chan, and Andrew Kyle Lampinen. 2025b. Just-in-time and distributed task representations in language models. In *Mechanistic Interpretability Workshop at NeurIPS 2025*.
- Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025c. From system 1 to system 2: A survey of reasoning large language models. *CoRR*.
- Ekdeep Singh Lubana, Can Rager, Sai Sumedh R Hindupur, Valerie Costa, Greta Tuckute, Oam Patel, Sonia Krishna Murthy, Thomas Fel, Daniel Wurgaft, Eric J Bigelow, and 1 others. 2025. Priors in time: Missing inductive biases for language model interpretability. *arXiv preprint arXiv:2511.01836*.
- Thang Luong, Edward Lockhart, and 1 others. 2025. Advanced version of gemini with deep think officially achieves gold-medal standard at the international mathematical olympiad. *Google DeepMind Blog*, 1.
- Uzay Macar, Paul C Bogdan, Senthoran Rajamanoharan, and Neel Nanda. 2025. Thought branches: Interpreting llm reasoning requires resampling. *arXiv preprint arXiv:2510.27484*.
- R Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D Hardy, and Thomas L Griffiths. 2024. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121.
- Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, and 1 others. 2021. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*.
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, and 1 others. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.
- Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. 2025a. Icl: In-context learning of representations. In *The Thirteenth International Conference on Learning Representations*.
- Core Francisco Park, Ekdeep Singh Lubana, and Hidenori Tanaka. 2025b. Competition dynamics shape algorithmic phases of in-context learning. In *The Thirteenth International Conference on Learning Representations*.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.
- Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. 2024. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- OpenAI Team. 2025. Introducing gpt-5.
- Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*.
- 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.
- Keyon Vafa, Peter G Chang, Ashesh Rambachan, and Sendhil Mullainathan. 2025. What has a foundation model found? using inductive bias to probe for world models. In *International Conference on Machine Learning*, pages 60727–60747. PMLR.
- Keyon Vafa, Justin Y Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. 2024. Evaluating the world model implicit in a generative model. *Advances in Neural Information Processing Systems*, 37:26941–26975.
- Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Lionel Wong, Katherine M Collins, Lance Ying, Cedegao E Zhang, Adrian Weller, Tobias Gerstenberg, Timothy O’Donnell, Alexander Lew, Jacob Andreas, Tyler Brooke-Wilson, and 1 others. 2025. Modeling open-world cognition as on-demand synthesis of probabilistic models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47.

Lionel Wong, Gabriel Grand, Alexander K Lew, Noah D Goodman, Vikash K Mansinghka, Jacob Andreas, and Joshua B Tenenbaum. 2023. From word models to world models: Translating from natural language to the probabilistic language of thought. *CoRR*.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2024. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. Association for Computational Linguistics.

Daniel Wurgaft, Ekdeep Singh Lubana, Core Francisco Park, Hidenori Tanaka, Gautam Reddy, and Noah D Goodman. 2025. In-context learning strategies emerge rationally. *arXiv preprint arXiv:2506.17859*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Itay Yona, Amir Sarid, Michael Karasik, and Yossi Gandelsman. 2025. In-context representation hijacking. *arXiv preprint arXiv:2512.03771*.

## A Further Examples of In-Context Representation Learning

In this section, we present three more examples of in-context representation learning over different topologies and different models. See Figures 10, 11, and 12 for a particularly diverse set of examples.

## B Optimal Context Lengths Per Model and Topology

Table 1 reports the optimal contexts lengths from the analysis of in-context representation learning described in Section 3.

## C Long Context Next Token Prediction

In this section, we present next-token prediction results for the next-token prediction task using the instruction prompt format, except we provide the all models with a consistent long context length of 1500. From Appendix B, we can see that this is longer than the previously analyzed context lengths for all but 3 model-topology combinations. From Figure 13, we see that this extended context length does not help performance.

## D Naive Solution for One Step Rules

This section describes a simple heuristic solution to adaptive world modeling tasks with a one-step rule, and presents the accuracies that this solution

Gemma-27b World Size 25 Line Walk Prefilled Representations

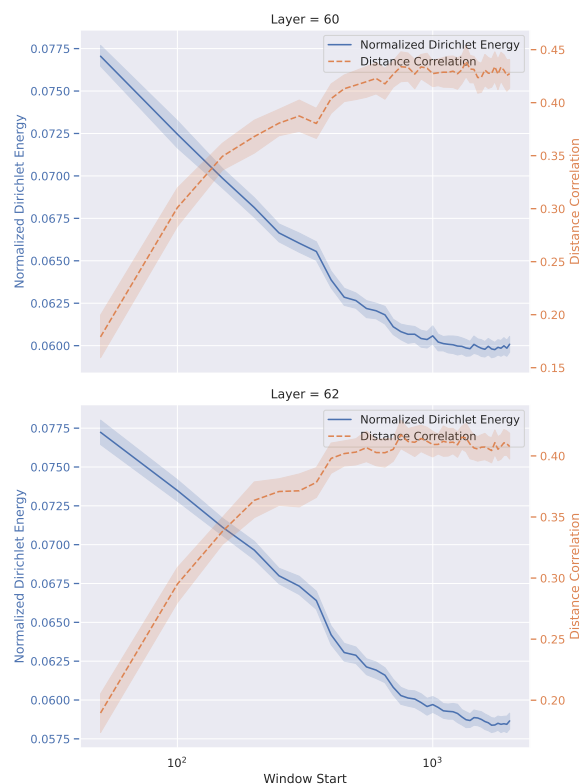


Figure 10: Gemma-3-27b-it in-context representation learning on a 25-length line topology.

Gemma-4b World Size 25 Grid Walk Prefilled Representations

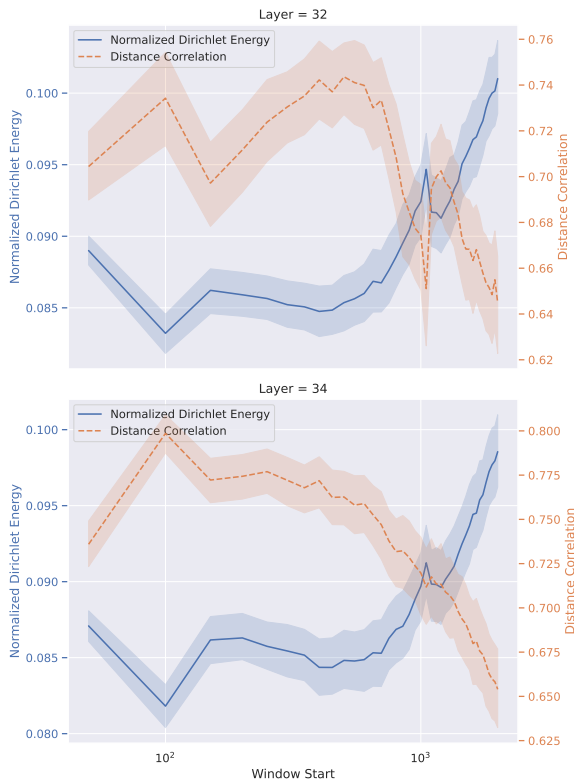


Figure 11: Gemma-3-4b-it in-context representation learning on a 5-by-5 grid topology. In this case, in-context representation learning is less well-behaved, though the distance correlation metric still increases over context.

OLMo-13b World Size 16 Line Walk Prefilled Representations

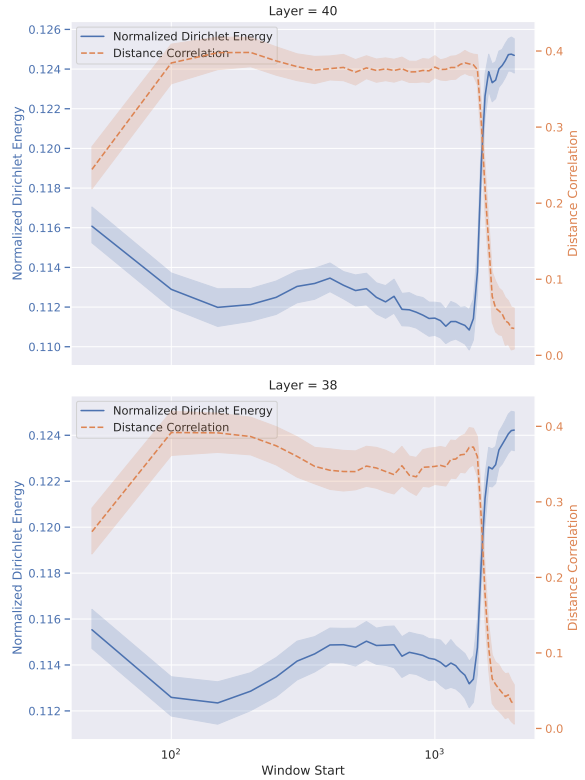


Figure 12: OLMo-2-13b in-context representation learning on a 16-length line topology. In this case, in-context representation learning occurs rapidly. This collapses much alter in the context.

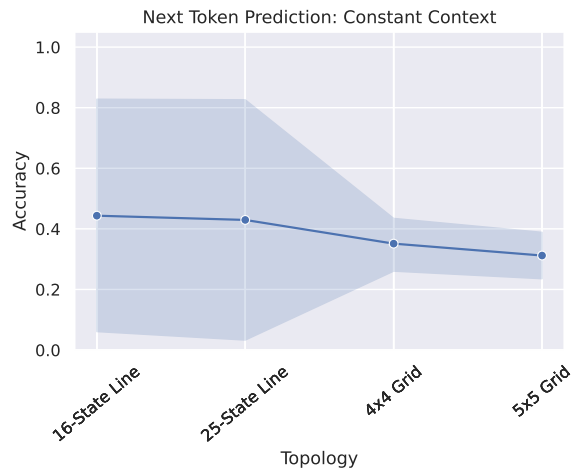


Figure 13: Next token prediction results with constant long context length.

Dimension	World Size	Baseline Acc.
1	16	53.3%
1	25	52.1%
2	16	33.3%
2	25	31.2%

Table 2: Naive baseline strategy accuracies.

Model	Prompt	Topology	Ctx. Len
Gemma-27b	Prefill	16-Line	1200
Gemma-27b	Prefill	25-Line	1350
Gemma-27b	Prefill	16-Grid	350
Gemma-27b	Prefill	25-Grid	650
Gemma-12b	Prefill	16-Line	450
Gemma-12b	Prefill	25-Line	500
Gemma-12b	Prefill	16-Grid	850
Gemma-12b	Prefill	25-Grid	1550
Gemma-4b	Prefill	16-Line	1300
Gemma-4b	Prefill	25-Line	1500
Gemma-4b	Prefill	16-Grid	300
Gemma-4b	Prefill	25-Grid	100
OLMo-13b	Prefill	16-Line	150
OLMo-13b	Prefill	25-Line	200
OLMo-13b	Prefill	16-Grid	100
OLMo-13b	Prefill	25-Grid	800
Gemma-27b	Instr.	16-Line	1150
Gemma-27b	Instr.	25-Line	1600
Gemma-27b	Instr.	16-Grid	400
Gemma-27b	Instr.	25-Grid	550
Gemma-12b	Instr.	16-Line	300
Gemma-12b	Instr.	25-Line	750
Gemma-12b	Instr.	16-Grid	650
Gemma-12b	Instr.	25-Grid	700
Gemma-4b	Instr.	16-Line	1600
Gemma-4b	Instr.	25-Line	1550
Gemma-4b	Instr.	16-Grid	200
Gemma-4b	Instr.	25-Grid	300
OLMo-13b	Instr.	16-Line	1400
OLMo-13b	Instr.	25-Line	1350
OLMo-13b	Instr.	16-Grid	250
OLMo-13b	Instr.	25-Grid	750

Table 1: Optimal context lengths for in-context representation learning

achieves. First, note that the input/output pairs given by the one-step rules (i.e., those that map  $S_i$  to  $S_{i+1}$  or  $S_{i,j}$  to  $S_{i+1,j}$ ) are attested in the random walk. Thus, a simple solution is to guess one of the attested adjacent states to the input state. However, not every state corresponds to a valid input state for a one-step rule (e.g., with zero-indexing  $S_{24}$  does not map to a valid state in a 25-state line, as  $S_{25}$  does not exist). Additionally, not every state has the same number of adjacent states. In the case of a line, the states at either end only have a single adjacent state. Randomly sampling one of the attested adjacent states for each possible query example gives the accuracies presented in Table 2.

## E Measuring Relative Dirichlet Energy

In Section 4, we present an analysis of the relative Dirichlet Energy between tokens in the random walk and tokens in the few-shot examples in an adaptive world modeling task. In this section, we provide details on this analysis.

First, we run 50 new adaptive world modeling prompts for each model and topology. We collect activations from the last 50 tokens in a random walk (or 50 adjacencies in a line topology). As described in Section 2, we compute average token representations for each token. We repeat this process separately for tokens that appear in the portion of the prompt containing few-shot examples. Notably, these two sets of tokens will be subsets of the tokens that occupy the state space (this is guaranteed for the tokens in the few-shot learning portion of the prompt, as several tokens are held-out by design). To account for this, we take the intersection of tokens occurring in both sets. Then, we compute normalized Dirichlet Energy as described in Section 2 for each of the sets independently, and report the ratio of these values.

To provide context for these results, we extract uncontextualized representations of the tokens present in the intersection of the two sets described above (corresponding to tokens in the last window of the random walk, or in the few-shot examples).

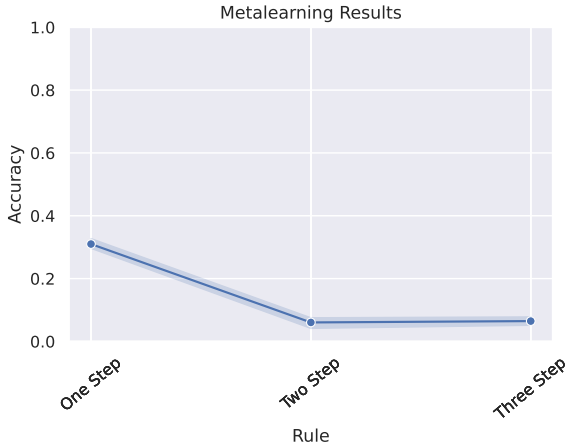


Figure 14: Metalearning prompt accuracy.

In this case, uncontextualized representations correspond to instances of these tokens that are at the very first position in the random walk. We compute and report the relative Dirichlet Energy of these uncontextualized tokens to the random walk tokens.

## F Metalearning AWM Prompt

In this section, we present the results of a metalearning version of the task, mentioned in Section 4. This variant consists of three task presentations, where each task presentation consists of a 500-length random walk over a 5-by-5 grid state space, followed by 3 few shot examples (all generated using the same rule). After these three task presentations, a query input is presented, and the model must infer the correct output. All task presentations are created using identical topologies, with the same tokens assigned to each state. These results are generated using a 5-by-5 grid topology. From Figure 14, we see that this approach yields poor performance.

## G Long Context Adaptive World Modeling

In this section, we present a similar analysis to Appendix C, except for the adaptive world modeling task using Gemma-3-27b-it. From Figure 15, we see that this does not improve performance.

## H Frontier Model Next-Token Prediction Results

In this section, we present results on the next-token prediction task using the frontier models described in Section 5. From Figure 16, we find that, aside from Gemini-2.5-Flash, models achieve good

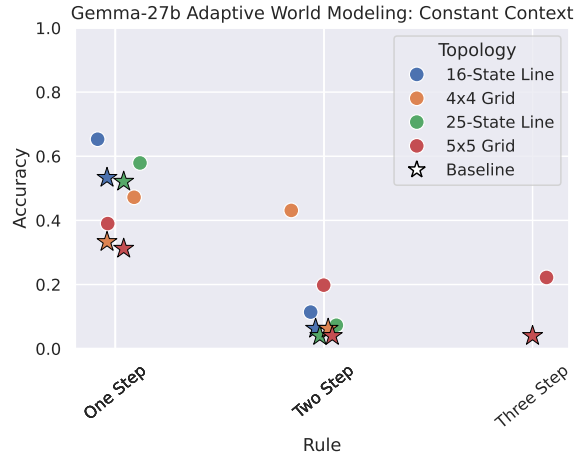


Figure 15: adaptive world modeling results with constant long context length, using gemma-3-27b-it.

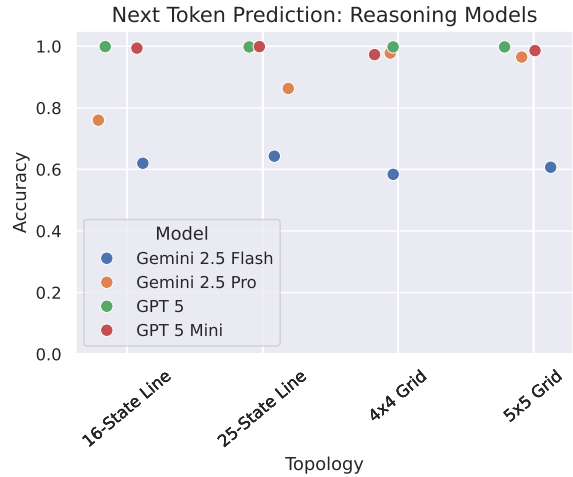


Figure 16: Frontier model next-token prediction performance.

performance at next-token prediction. Surprisingly, this is especially true for grid topologies.

## I Ablating Thinking

In this section, we ablate the Gemini model’s ability to use thinking tokens and analyze the impact on next-token prediction and adaptive world modeling. For Gemini-2.5-Flash, we can entirely ablate the model’s ability to produce thinking tokens. However, for Gemini-2.5-Pro, we can only set the thinking budget to a minimum of 128 tokens. For Pro, we see that limiting thinking to 128 tokens often hurts performance on both tasks (See Figures 17 and 18). For Flash, however, we see mixed results. In many cases (and especially for next-token prediction on line topologies), ablating thinking drastically *improves* performance (See

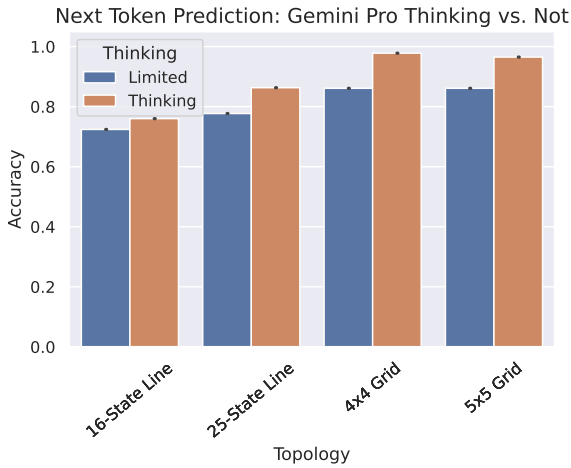


Figure 17: Gemini-2.5-Pro Thinking ablation for next-token prediction.

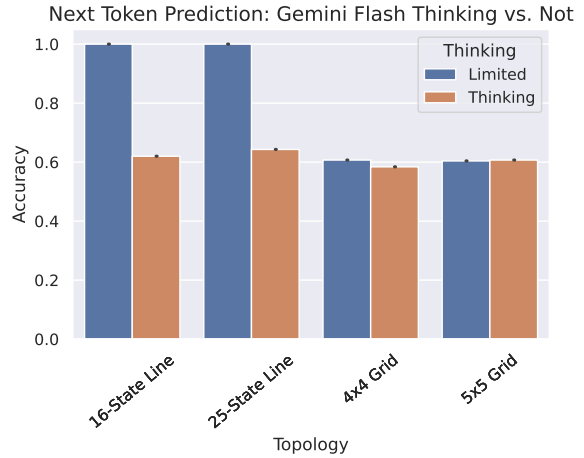


Figure 19: Gemini-2.5-Flash Thinking ablation for next-token prediction.

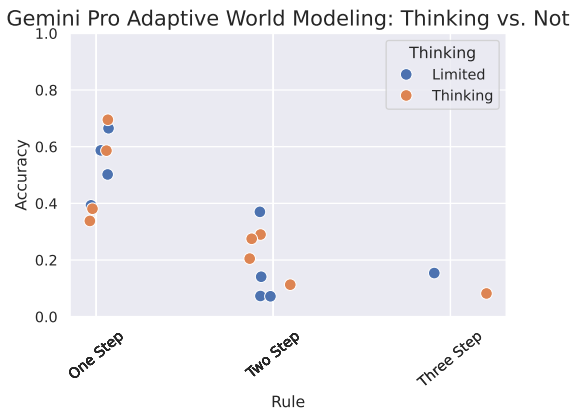


Figure 18: Gemini-2.5-Pro Thinking ablation for adaptive world modeling.

Figures 19 and 20). More work must be done to understand the implications of these results. One speculative hypothesis is that entirely ablating thinking forces the model to rely on forward pass latent representations, rather than (potentially faulty) externalized thinking traces. These latent representations have very likely undergone in-context representation learning, and (perhaps) the model has learned to flexibly use these representations in this very specific, limited context.

## J The Impact of Hints

In this section, we analyze frontier model performance when we provide strong clues about the underlying state space topology. In particular, here we include information about the configuration of the grid (4-by-4 or 5-by-5) in the prompt, and also mention that the few-shot examples map states at one position to states at another position. Even

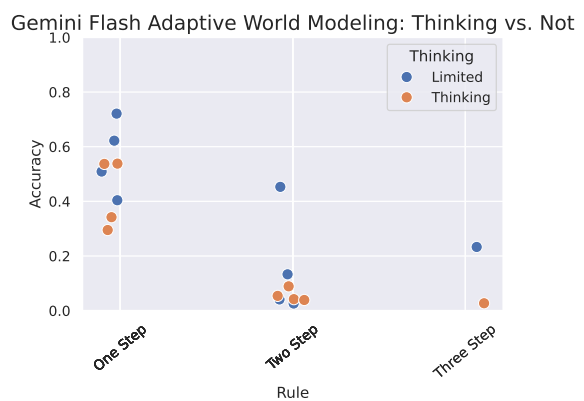


Figure 20: Gemini-2.5-Flash Thinking ablation for adaptive world modeling.

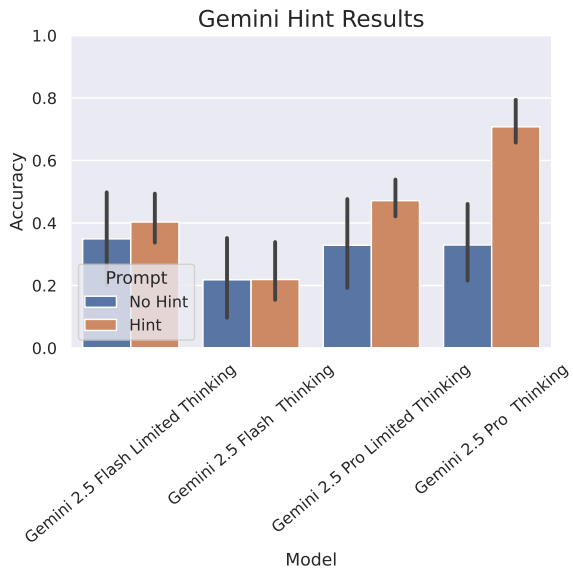


Figure 21: Adaptive world modeling performance on frontier models with and without hints on 2-dimensional topologies.

with these very strong hints, models still do not achieve perfect adaptive world modeling accuracy (See Figure 21).