WORLD MODELING IN TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Behavioral failures can make a transformer appear to lack a world model even when it has learned faithful representations of its environment. We demonstrate this in TaxiGPT, a transformer trained on random walks through Manhattan whose failures have been interpreted as evidence of an incoherent internal map. Through mechanistic analysis and causal interventions, we show that the model represents intersections and streets, tracks its position, and uses a goal compass to navigate. We trace its failures to interference between superposed intersection features, which disrupts localization within the internal map. Affordance packing, which groups representations of intersections with the same legal moves, helps limit the consequences of these errors. Finally, we propose mechanistic indicators that we use to compare models and show that world-modeling capacities emerge at different stages of training. Our findings motivate a shift from asking whether a model *has a world model* to mechanistically studying its *world modeling*: the interacting capacities through which it represents its environment and uses those representations to guide behavior.

1 INTRODUCTION

Among the more exciting promises of transformers is the prospect that entire "world models" might emerge inside them just from training on sequences of data in a domain. Yet assessing whether a transformer has recovered a faithful world model is more challenging than it first seems. A case in point is what we dub "TaxiGPT": Vafa et al. (2024) trained GPT-2-style transformers from scratch on taxi rides through Manhattan to see whether they would recover a faithful street map. The models were trained to predict the taxi's next turn given a sequence of tokens encoding the taxi's origin, destination, and preceding turns ("N NW NE E SW..."). The most accurate of these TaxiGPT models, trained on random walks, was able to output legal turns 99% of the time. But it operated in an environment in which even a random guess had a non-zero chance of picking out a legal move. And when Vafa et al. sought to reconstruct the maps implicit in the model's outputs, the map looked more like spaghetti than like the street layout of Manhattan. The quality of outputs also deteriorated sharply when the researchers intervened to force the taxi away from its destination three quarters of the time. Vafa et al. concluded that their transformer was "very far from recovering the true street map of New York City" (Vafa et al., 2024, p. 2).

The difficulty, however, is that a model's behavior underdetermines the mechanisms that produced it. Reconstructing the map implicit in its outputs cannot distinguish whether the model actually contains an incoherent map, or whether it is the mechanisms that locate and guide the model within the map that misfire.

Our mechanistic analysis of the TaxiGPT model trained on random walks reveals that the model in fact harbors a *highly faithful internal map* of Manhattan. The model accurately represents the intersections and the streets connecting them. TaxiGPT keeps track of its location on the map using a running state informed by a rolling window of previous positions. Building on causal investigations of learned world representations (Li et al., 2023a; Spies et al., 2025), we use targeted interventions to establish that the model indeed exploits this map to navigate. Its representation of the current intersection favors moves that are legal at that intersection.

To guide its selection between legal moves, it uses a *goal compass* that encodes the destination's direction and favors legal moves leading toward it. Ablating the compass preserves legal moves but severely impairs the model's ability to reach distant destinations. This shows that simply identifying

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

a faithful internal map does not suffice to explain the model's navigational abilities. Mastering legal turns within the street network and traveling toward a destination are distinct achievements calling for different internal resources. There is more to world modeling than map recovery.

The question then becomes how a model using such a faithful map can nevertheless produce routes that suggest an incoherent one. We find that TaxiGPT stores intersection representations in superposition. Thousands of intersection features occupy a small subspace. Out of distribution, the signal specifying the current position weakens and interference increases, allowing incorrect intersection features to promote illegal moves. We trace the failures observed by Vafa et al. (2024) to this phenomenon. Strengthening the correct position signal or suppressing interference substantially restores the model's performance.

We further show that the model *organizes this superposition by affordance*: intersections with the same set of legal next moves tend to have nearby representations. This *affordance packing* makes many localization slips benign with respect to immediate move legality: a move that is legal at the mistaken intersection is also legal at the true one. The rolling window of recent positions can then help restore correct localization before the error propagates. Vafa et al. (2025) also found evidence that sequence models tend to group states with the same permissible next moves. Prieto et al. (2026) showed that overlapping representations of correlated features can helpfully reinforce one another. Our findings illustrate a different benefit of affordance packing: it mitigates the damage caused by localization slips.

These findings suggest that instead of looking for a single, self-contained "world model" at one layer, we should look for multiple complementary *world modeling* strategies: the mechanisms through which a transformer represents environmental structure, tracks its situation within it, and uses that information to guide prediction or action.

Finally, this decomposition yields *mechanistic indicators of world-modeling capacities* that we use to compare architectures, datasets, and training objectives, and to track the emergence of different world-modeling capacities through training. This reveals that they do not emerge in lockstep: in the run studied, legal-move and goal-direction representations mature before precise localization, while the position code expands and becomes increasingly organized by affordance.

These results motivate a shift from identifying a *world model* to explaining the processes of *world modeling*. A model is something a system *has*; modeling is something it *does*, often in several ways at once, and with uneven success. Learning the structure of an environment and navigating it reliably are distinct achievements, which is why a faithful map can coexist with unreliable navigation. Neither behavioral success nor representational fidelity on its own suffices to settle whether and how a transformer models its world.

2 World modeling in TaxiGPT

2.1 A highly faithful internal map, stored in superposition

We inspect GPT2-XL ($48\text{L} \times 1,600$ dims) trained on random walks over the map of Manhattan.

Intersections are encoded. We test two methods for extracting representations of the intersections, or nodes, of the Manhattan map: linear probes and diff-means (Marks & Tegmark, 2023; Li et al., 2023b). Diff-means features score best. They decode most accurately at layer 18, where 99.6% of intersections are decoded with at least 90% accuracy on held-out data (App. A.1). We next test whether these representations play a causal role in the model's predictions through a minimal teleportation edit: we shift the taxi's encoded position to an intersection next to the goal and check whether its next greedy prediction reaches the goal from that position. Using the layer 11 direction, this succeeds on at least one eligible scene for 99.3% of testable nodes (App. A.2). Finally, the intersection features encode Manhattan's spatial layout (App. A.3).

Streets are encoded. First, the intersection features themselves encode which moves are legal from that point on the map. Applying the logit lens at layer 31 with a 1% probability threshold recovers the exact legal-move set for 99.4% of intersection features (App. B.1). But this does not yet tell us whether the model encodes which intersections are connected by each street. Two complementary tests provide evidence that the model encodes these too. The *steering test* asks whether an intersection

feature and a move activate the feature of the correct next intersection. We inject an intersection feature, feed a move, and measure which intersection feature grows most afterward. We average the other prompt states to avoid tying the test to a particular route. The correct next intersection's feature grows most for 76.6% of tested streets and ranks among the top five for 93.3% (Fig. 1a). The *probing test* asks whether these transitions can also be recovered linearly from the model's residual-stream state during rides. We train a separate linear probe for each move to predict the next intersection's feature direction, then identify the intersection feature closest to the probe's output. This recovers the correct successor for 89.2% of legal state–move queries at held-out intersections (App. B.2).

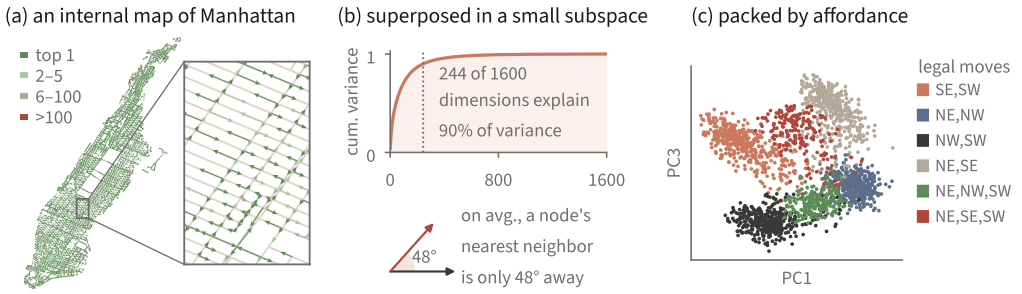


Fig. 1: **The map encoded inside TaxiGPT.** (a) The Manhattan map recovered by the street steering test: we inject an intersection feature, feed a legal move, and measure which intersection feature grows most afterward (color shows the correct next intersection's rank); (b) the intersection features are superposed in a small subspace of the residual stream; (c) and are packed by affordance: intersections with the same legal next moves cluster together (main affordance classes shown).

The map is stored in superposition and packed by affordance. The 4,516 intersection directions of the position code occupy a small subspace of the 1,600-dimensional residual stream: 90% of their variance lies in 244 dimensions. The angle between an intersection direction and its nearest neighbor averages 48.1° (Fig. 1b). This superposition is organized by affordance. We group intersections by their sets of legal next moves, yielding 104 groups, and compute the mean feature vector for each group. For 91% of intersections, the feature vector has higher cosine similarity to its own group's mean than to any other group's mean. Thus, intersections offering the same legal moves tend to have aligned features (Fig. 1c). Intersections that are close on the map also tend to have aligned features, though this relationship is weaker (App. A.4).

2.2 Localization and Navigation

Two world modeling capacities are linked to the internal map: the model *localizes* itself on the map (tracks where it is) and *navigates* (works out which way to go from there).

Localization. TaxiGPT localizes by reading the active intersection features in the residual streams of several past positions, which we call its *look-back window*. It thus maintains a running position estimate rather than recomputing its location from scratch using the sequence of moves (Fig. 2a; App. C.1).

Navigation. Two complementary mechanisms shape the model's move predictions (Fig. 2b). The active intersection feature increases the logits of legal moves and decreases those of illegal ones. The *goal compass* increases the logits of moves toward the goal and decreases those of moves away from it. This compass is a circular representation in the residual stream (similar to the circular features identified by Engels et al., 2025; Wurgaft et al., 2026) that encodes the direction from the current intersection to the goal. To identify it, we group rides into 16 bins by goal bearing, extract a difference-in-means direction for each bin, and combine these directions to obtain the compass plane. Goal bearing is decoded best at layer 16, with a median angular error of 18.1° on held-out rides. Two interventions establish the compass's causal role: steering it makes the model follow the selected direction over 12 moves, with a median angular deviation of 18.1°, while ablating it preserves move legality but severely impairs the model's ability to reach distant goals (App. C.2).

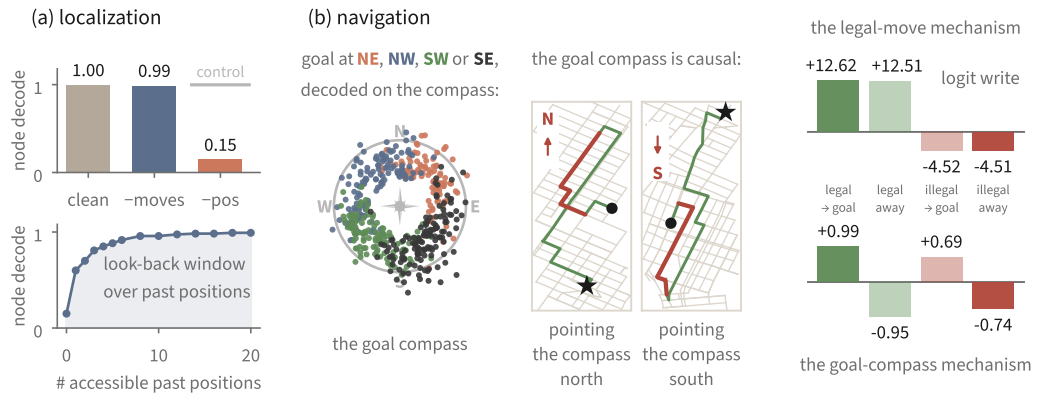


Fig. 2: **Using the map.** (a) *Localization*: the model carries a running position, read from a look-back window of past positions. Erasing past positions hurts current-intersection decode while erasing the past moves does not (measured on rides of 27–34 moves). (b) *Navigation*: a goal compass encodes the bearing to the goal. It is also causal: clamping it north or south makes the taxi follow that direction. Two mechanisms write into logits: a legal-move mechanism (activated by the intersection feature) favors legal moves while the goal-compass mechanism favors goalward moves.

Other mechanisms. The model uses an *at-goal* feature to decide when to stop (App. D.1). It also uses what we call a *commit-to-goal* feature. Because TaxiGPT is trained on random walks, it reproduces their statistics: routes wander rather than head straight for the goal, and often overshoot it, looping back before stopping. We find a single direction at layer 16 that controls this trade-off, dialing the model between extreme random-walk behavior and shortest-path behavior (App. D.2).

These findings suggest that TaxiGPT satisfies two conditions for world representation discussed in the philosophical literature: *structural isomorphism*, supported by the encoding of intersections and streets, and *exploitation*, supported by the causal teleportation test (Shea, 2014; Williams, 2026).

3 World Modeling in Superposition and Resulting Failure Modes

We now explain why a model using such a faithful map can nevertheless produce the behavioral failures reported by Vafa et al. We inspect their stress test, detour test, and compression metric. We show that all three push the model into an out-of-distribution regime in which the current-position write grows weaker and the noise in the position subspace higher. Because the map is stored in superposition, these perturbations can activate a wrong node, making the next move illegal; though the affordance packing limits how often this happens (Fig. 3). We now look at each test in turn.

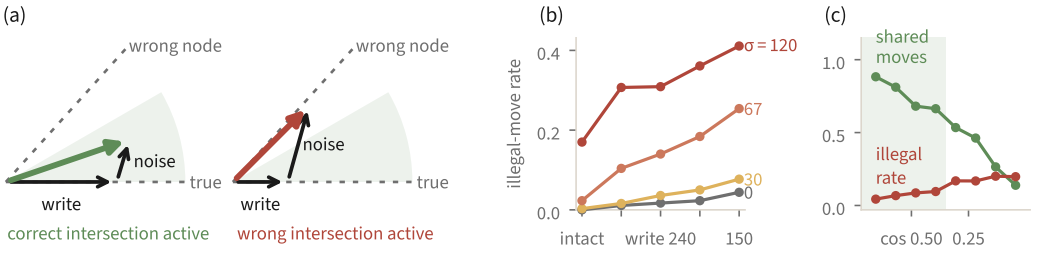


Fig. 3: **A weak write and noise in the position code are enough to cause off-graph moves.** (a) When the true-position write is weak, noise can make a wrong intersection feature most active. (b) We causally reproduce this phenomenon by weakening the write and adding noise to correctly localized states, causing the model to emit illegal moves (700 stress rides with initially legal top moves, depth ≥ 60; edits at layer 18). (c) The farther the wrong feature lies from the true direction, the fewer legal moves the two intersections share and the more likely an illegal move becomes. The nearby region protected by affordance packing is shaded green in (a) and (c).

3.1 THE STRESS TEST

We use *stress test* to refer to the evaluation underlying Vafa et al.'s reconstruction of the map implicit in TaxiGPT's outputs. The model generates rides between sampled origin–destination pairs (with temperature 1), and the resulting illegal moves are overlaid on Manhattan's true street map. These sampled pairs place the model in a genuinely out-of-distribution regime: they are a median of 32 moves apart, whereas training rides start 9 moves from their goal (App. E.1). Because TaxiGPT replicates the meandering of its training distribution, the goal is often unreachable within its budget of 99 moves (after which it has no more trained position encodings). The model still generalizes quite well on the stress test: it reaches the goal on 81% of pairs and the taxi's current intersection can be decoded from the model's internal activations with 99% accuracy. But it does take an off-graph move on 8.5% of the rides. These failures stem from a weak write plus noise in position space, which can make a wrong node the most active and cause the model to emit an illegal move (Fig. 4). Given the amplitude of the perturbation, we can distinguish four categories of failure mode that lead to an illegal move (Fig. 5).

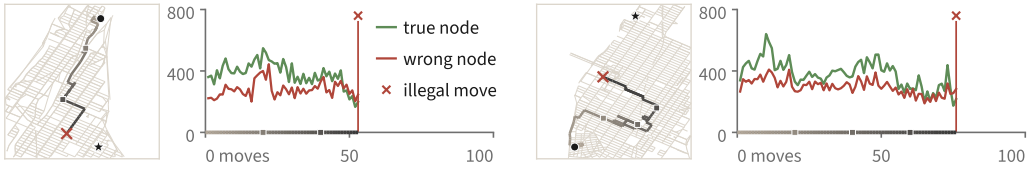


Fig. 4: **Two stress rides where a wrong node becomes most active, leading to an illegal move.** The shared fluctuations reflect how superposition works: when the true node is written more strongly, wrong nodes with superposed features also become more active.

Recovering superposition slips. *Superposition slips*, in which an incorrect intersection feature becomes most active, are mostly benign. In benign cases, corruption of the position code is mild, combining a weak current-position write (median 360; Fig. 5) with modest noise (median 56). The activated feature is close in direction to the correct one (median cosine similarity 0.65), within the zone protected by affordance packing (median shared affordance 1.0), so the next move remains legal. The look-back window over past positions then allows the model to *recover* from the slip (App. E.3).

Fatal slip (25%). When the signal representing the true intersection weakens further (median strength 244), a less closely aligned intersection representation can become active (0.38). The mistaken intersection shares fewer legal moves with the true one (0.33), so a move that is legal there may be illegal at the taxi's actual position. Removing the wrong node activation (the component orthogonal to the true node), removing the position noise altogether (at L18), or strengthening the write all cause the illegal probability mass to drop significantly (0.55 / 0.80 / 0.56).

Silent slip (24%). Even when the correct node is the most active feature, wrong co-active nodes sometimes still leak illegal moves into the logits through their affordances. They are almost entirely responsible: removing the position noise drops the illegal probability by 0.90.

Full corruption (29%). When the write weakens further (median 171), the corruption leaves the superposition regime: the strongest competing node has little overlap with the true node (cosine 0.08, shared affordance 0). Removing the top node therefore helps less (0.34); clearing all position noise (0.63) or rebuilding the write (0.75) gives greater recovery.

Give-up slips (16%). Deep in a ride, with the goal still far away, a give-up feature becomes active and promotes stopping (App. E.4). In this regime, the residual is enlarged and noise in the position code can still produce slips. The write is stronger than in full corruption (median 296), but noise is also higher (91). Clearing this noise reduces illegal probability mass by 0.74.

These four categories account for 93.5% of illegal moves. Of the rest, 6% are low-mass unlucky draws: the model places under 0.001 total probability on off-graph moves (our marginal threshold), but temperature-one sampling drew one anyway. The remaining 0.5% fall outside these categories (App. E.3).

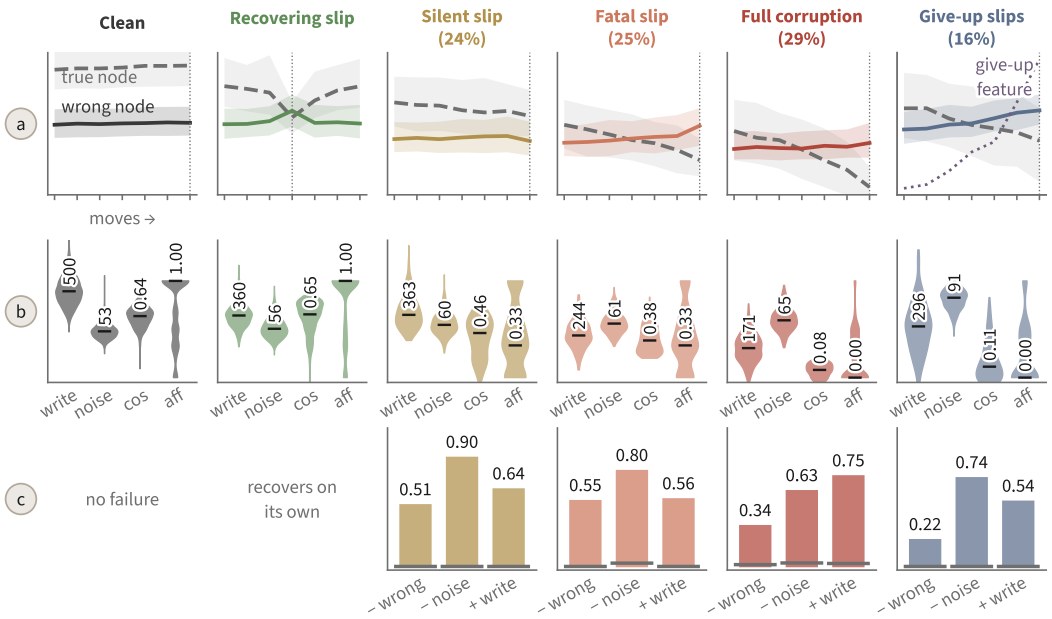


Fig. 5: **TaxiGPT's failure modes.** (a) The first row shows how true-node (dashed) and wrong-node (solid) activations change over successive moves, comparing clean rides, recovering slips and four failure modes. (b) In these failure modes, weaker writes (write ↓) and stronger noise (noise ↑) let the wrong node lie at a larger angle (cos ↓) and share fewer legal moves (aff ↓), making illegal moves more likely. (c) As a causal check, we remove the wrong-node activation (− wrong), remove position noise (− noise), or strengthen the true-position write (+ write), and show the fraction of illegal-move probability removed (gray marks: controls with edits of the same size along random directions).

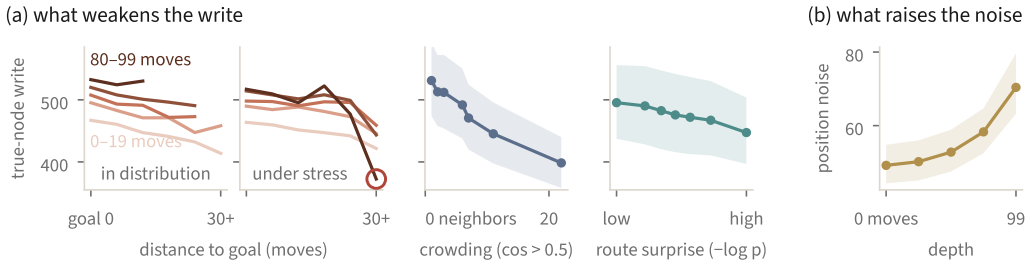


Fig. 6: **Factors affecting write strength and noise.** (a) In distribution, the write (measured at layer 18) falls with distance to the goal but rises with depth. Under stress the pattern is broadly the same, except in the out-of-distribution deep-and-far regime (red ring), where the write drops sharply. The write is also weaker when the current intersection's representation overlaps with many others (crowding), and when the route is less predictable (route surprise). (b) Noise grows with one factor: depth. (Lines are medians; bands are the interquartile spread.)

What shapes write strength and noise? We observe that the write weakens with distance to the goal, node superposition, and route surprise, and strengthens with depth (Fig. 6). However, the stress test introduces a regime that training almost never visits: *deep-and-far*, where the model has taken at least 60 moves and remains at least 20 moves from the goal (0.1% of in-distribution states against 5.4% under stress). There, depth and distance compound unexpectedly and the write becomes sharply weaker. Note that the distance-to-goal effect seems to be intentional: swapping only the destination token in stress-ride states weakens the write in 87% of cases (median decrease: 59) (App. E.2). The noise grows with depth.

The stress test thus reveals a *misrecruitment* of the position code, driven largely by interference between superposed intersection features. The failing world modeling capacity is primarily that of

localization: the representations remain available, but the model struggles to recruit them correctly under challenging conditions.

3.2 THE DETOUR TEST AND THE COMPRESSION METRIC

The detour test (App. F). At each move, the adversarial detour test overrides greedy decoding with probability 0.75 to force the least-likely legal move. Once the remaining budget just suffices to reach the goal, forcing stops and the model continues greedily. The model emits an illegal move on 25.6% of rides. We find that individual forced moves do not disrupt the internal representations more than non-forced ones. However, because the least-likely legal move almost always points away from the goal, repeated forcing pushes the ride into the deep-and-far regime and along routes highly unlikely under the learned distribution. These conditions significantly weaken the current-position write. The same four failure modes appear as in the stress test, with a shift toward full corruption, and the same interventions improve move legality.

The compression metric (App. G). Vafa's compression metric takes two same-length routes (prefixes) that end at the same intersection with the same goal, samples 30 continuations (suffixes) from one, and checks whether they remain likely under the other (probability $> \epsilon$). The logic is that prefixes encoding the same state should support the same continuations. However, our results point to the difficulty of the full ride formed by the prefix + suffix, rather than a localization mismatch at the end of the prefixes. Indeed, both prefixes decode to the correct shared intersection in all 146 pairs we inspect, including the 94 whose continuations fail the test. What happens instead is that the prefixes and suffixes form challenging rides that again enter out-of-distribution regimes, including deep-and-far states, and exhibit the same four failure modes. To test whether compression tracks these difficulties, we lengthen the suffixes by moving the goal farther away while keeping the prefix routes fixed. Compression falls from 0.983 to 0.167 and illegal moves become more frequent, although both prefixes still decode to the correct shared intersection in 99.8% of conditions. Across the seven distance bands, compression correlates strongly with illegal-move rate ($r = -0.957$), suggesting that compression is sensitive to the same localization failures as the stress and detour tests.

Localization failures thus contribute to TaxiGPT's poor performance on the detour test and compression metric. As a final confirmation, we continuously reinforce the correct position at L11 during all tests (adding the correct intersection's diff-means vector after each move). This improves stress-test legality (91.5% → 97.3%), detour success (63.1% → 71.9%), and compression (0.524 → 0.691; App. G). This shows that the position code remains usable and effective; the difficulty lies in recruiting it reliably.

4 MECHANISTIC INDICATORS OF WORLD MODELING

We re-use the probing techniques developed for this case study as mechanistic indicators. This enables us to compare world-modeling capacities across models, training data, and training regimes (Table 1) and to track their emergence during training (Fig. 7).

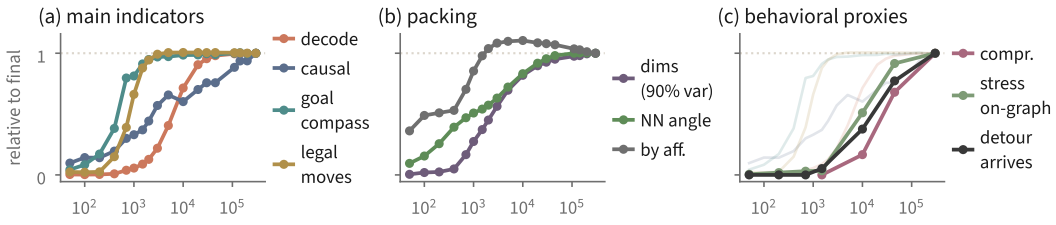


Fig. 7: **Emergence of world-modeling capacities through training** (RW·NTP, 384d). Navigation capacities (compass and legality) emerge first, localization (decode and causal) second (a). The position code expands (nearest-neighbor angles grow); affordance packing rises early, then relaxes as intersections differentiate (b). Mechanistic indicators can be compared with behavioral proxies (c). Values are normalized to the final checkpoint. (Protocol and controls in App. H.2.)

Mechanistic indicators give a more fine-grained diagnostic than behavioral proxies. Behavioral scores alone cannot distinguish which world-modeling capacities are present in a model and which

Table 1: **Comparing world modeling capacities across architectures, data and training regimes.** For each indicator, we select the layer where it scores best in its corresponding sweep. Rows are dataset · objective: `SP` shortest paths, `NSP` noisy shortest paths, `RW` random walks; `NTP` next-token prediction, `NextLat` next-latent prediction (Teoh et al., 2026). (Protocol, controls, and training budgets in App. H.1.)

	Map / localization					**Navigation**		**Behavior**		
	decode	causal	streets probe	super-position	by afford.?	legal moves	goal compass	stress test	detour test	compr.
	% int $\geq .9$	telep $>50\%$	held out	angle / dims	own class	% int.	steer err	on graph	success	score
`SP·NTP`	12.7%	14.9%	23.5%	35.9°	41.0%	18.1%	25.8°			
12L×768d×12h	L6	L10	L7	162d, L6	L6	L10	L12	71.7%	0.0%	.101
`NSP·NTP`	17.1%	16.9%	12.2%	38.0°	41.1%	22.3%	23.2°			
48L×1600d×25h	L43	L46	L36	251d, L43	L43	L44	L47	74.7%	0.2%	.054
`RW·NTP`	99.6%	94.0%	89.2%	48.1°	91.4%	99.4%	18.1°			
48L×1600d×25h	L18	L11	L15	244d, L18	L18	L35	L15	91.5%	63.1%	.524
`RW·NTP`	99.7%	88.5%	81.9%	47.1°	88.2%	96.9%	17.3°			
48L×384d×8h	L40	L31	L39	210d, L40	L40	L44	L43	96.7%	77.5%	.523
`RW·NextLat`	99.9%	87.1%	94.8%	42.2°	84.7%	99.0%	17.6°			
48L×384d×8h	L36	L27	L44	183d, L36	L36	L44	L37	97.3%	78.5%	.556

are not. Mechanistic indicators can: Table 1 shows, for example, that SP and NSP learn a causal goal compass despite weak intersection decoding and street probing. We take the decode indicator to be particularly useful because it tests both whether the model has representations that distinguish intersections and whether it reliably recruits the correct representation during inference to localize. It reliably separates the better- and worse-performing models (Table 1), follows a smooth sigmoid during training, and closely tracks stress and detour performance (Fig. 7). These results support its use as an indicator in other world-modeling tasks, particularly when the decoded representations are also shown to be causally used, as we establish here for the diff-means features.

World-modeling capacities emerge at different stages of training. In the run studied, navigation capacities (legal moves and goal compass) mature before localization capacities (decode and causal) (Fig. 7). The diff-means intersection features encode legality before they reliably distinguish intersections, suggesting that the model first groups intersections by legal-move set, then differentiates them within each group. The goal compass provides a sense of direction before the model reliably distinguishes the map's individual intersections and recruits their representations to guide its moves. These early spatial capacities may provide a foundation for learning the map's precise relational structure. Stress and detour performance improves as this more precise structural understanding develops. More generally, this suggests that we should expect world-modeling capacities to sometimes develop unevenly, with some supporting the emergence of others.

Better navigation need not imply a better internal map. Smaller models achieve better behavioral scores, yet the large RW model scores comparably on the mechanistic indicators and better on the causal indicator. The behavioral gap appears to reflect differences in reliably using these representations: the small RW·NTP model slips 3× less and RW·NextLat 6× less (App. H.1). This also reinforces the distinction between learning a faithful map and reliably locating oneself within it.

Data, architecture, and training objective shape world modeling. First, data appears to be the most important lever: across all architectures, RW models have far better world-modeling representations (Table 1). Given that the recorded failure modes emerge out of distribution, the data could likely still be improved considerably (for instance by training on the deep-and-far regime). Second, the architecture constrains world modeling: smaller RW models place their intersection representations at later layers ($L27$–$L40$ versus $L11$–$L18$), suggesting that they compensate for a narrower residual stream by spreading the necessary computation over more layers. Third, the training objective can encourage the emergence of world-modeling capacities. NextLat, which adds an objective of predicting the next latent state, achieves the best mechanistic scores and, notably, encodes streets best. This fits the objective: streets determine which intersection comes next after a move, so learning to predict the next latent state should encourage the model to encode them.

Affordance packing supports both early prediction and later robust world modeling. During training, the position code expands within the residual stream alongside improvements in world modeling capacities (Fig. 7). Better models likewise show less superposed features (Table 1). The model learns to predict legal moves before it can reliably distinguish individual intersections, and this early improvement accompanies increasing affordance packing. Affordance packing thus first serves as an initial way to lower prediction loss. As training progresses, individual intersections become distinguishable while the grouping largely persists. This is coherent with our failure analysis: affordance packing helps protect against superposition interference; insofar as this protection lowers training loss, it provides a further pressure to preserve (and potentially reinforce) the packing. Our results thus show that an affordance bias (viewed critically by Vafa et al. 2025) can coexist with real world modeling, and even help make it more robust.

5 Related work

World representations and their use. Behavioral evaluations reveal limitations in adaptive planning (Momennejad et al., 2023), state-consistent prediction (Vafa et al., 2024), and transfer across tasks sharing the same underlying structure (Vafa et al., 2025). Mechanistic interpretability studies have identified world representations and state-tracking mechanisms in Othello (Li et al., 2023a; Nanda et al., 2023), chess (Karvonen, 2024), maze-solving transformers (Ivanitskiy et al., 2024; Spies et al., 2025), permutation tasks (Li, Guo, and Andreas, 2025a; Zhang et al., 2025), and spatial language tasks (Tehenan et al., 2025; Xia et al., 2026). Recent work formalizes the distinction between representing and using world structure (Li, Viégas, and Wattenberg, 2025b). Lepori et al. (2026) find that models can learn representations in context yet sometimes fail to use them when needed. Our analysis connects these approaches by showing how a causally used, faithful map can nevertheless produce behavior that suggests an incoherent one.

Packing in superposition. Interference between superposed features can cause errors (e.g., Stevinson et al., 2025), and smart packing can limit the damage. Known strategies include antipodal packing of features that do not co-occur (Elhage et al., 2022), correlated packing of features that do (Elhage et al., 2022), which can be constructive (Prieto et al., 2026), and hierarchical packing (Park et al., 2025; Bussmann et al., 2025). We find affordance packing, where states that permit the same next actions are placed close together, limiting the behavioral consequences of confusing their representations.

6 Conclusion

We investigate a model operating in a world with a finite number of states and deterministic transitions. What additional capacities are needed for world modeling in more complex settings is a question for future research. Nevertheless, TaxiGPT offers lessons about both the challenges of world modeling and how we should assess it.

Superposition poses a challenge to reliable world modeling: interference between representations can cause the model to apply its correct representations of the environment to the wrong situation. But training may also find ways to limit this damage. Affordance packing helps preserve move legality during localization slips: intersections with the same legal moves tend to have nearby representations, so confusing them need not produce an illegal move. The look-back window of recent positions can then support recovery.

Mechanistic interpretability helps distinguish explanations that behavior alone leaves unresolved. A navigation error can stem from shortcomings in representing the environment, locating oneself within it, determining which moves are legal, or selecting those legal moves that help one progress toward a destination. By identifying how world-modeling capacities are implemented and interact, mechanistic analysis allows us to trace where things go wrong beyond behavioral tests.

More broadly, our findings motivate a shift from asking whether a model has a world model to examining its *world modeling*. This means asking what environmental structures a model has learned, which capacities recruit these structures to guide behavior, and under what conditions those capacities work together reliably. Learning to accurately map out an environment is but a start; reliable world modeling requires learning how to make good use of that map.

AI USE STATEMENT

We used LLMs throughout the research and writing process, including experiment design, implementing experiments (coding), figure preparation, and manuscript revision. In particular, rapid implementation of preliminary experiments let us test a wider range of hypotheses and identify promising signals for closer investigation. We also continuously asked LLMs to identify errors in our claims and code. We did not use LLMs to propose hypotheses, develop the conceptual framework, interpret results, generate data, or for translation; dataset cleaning or reformatting, mathematical claims and proofs, and qualitative analysis are not applicable to this work. All AI-assisted code, results, and text were reviewed by the authors, and the code is publicly released for verification. We take responsibility for the final code, results, and manuscript.

REPRODUCIBILITY STATEMENT

Experimental details are provided in the appendix. Code and instructions for running the experiments are available at https://anonymous.4open.science/r/taxiGPT-ED9B/.

REFERENCES

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In *International Conference on Machine Learning*, 2025. arXiv:2503.17547.

Nelson Elhage, Tristan Hume, Catherine Olsson, et al. Toy models of superposition. *Transformer Circuits Thread*, 2022. arXiv:2209.10652.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *International Conference on Learning Representations*, 2025. arXiv:2405.14860.

Michael I. Ivanitskiy, Alex F. Spies, Tilman Räuker, et al. Structured world representations in maze-solving transformers. *arXiv preprint arXiv:2312.02566*, 2024.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *Conference on Language Modeling*, 2024. arXiv:2403.15498.

Michael A. Lepori, Tal Linzen, Ann Yuan, and Katja Filippova. Language models struggle to use representations learned in-context. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14841–14857, 2026. doi: 10.18653/v1/2026.acl-long.676. URL https://aclanthology.org/2026.acl-long.676/.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (how) do language models track state? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34429–34452, 2025a. URL https://proceedings.mlr.press/v267/li25r.html.

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*, 2023a. arXiv:2210.13382.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023b.

Kenneth Li, Fernanda Viégas, and Martin Wattenberg. What does it mean for a neural network to learn a "world model"? *arXiv preprint arXiv:2507.21513*, 2025b.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Ida Momennejad, Hosein Hasanbeig, Felipe Vieira, et al. Evaluating cognitive maps and planning in large language models with CogEval. In *Advances in Neural Information Processing Systems*, 2023.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *BlackboxNLP*, 2023. arXiv:2309.00941.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *International Conference on Learning Representations*, 2025. arXiv:2406.01506.

Lucas Prieto, Melody Stevinson, Melih Barsbey, Tolga Birdal, and Pedro A. M. Mediano. From data statistics to feature geometry: How correlations shape superposition. In *International Conference on Learning Representations*, 2026. arXiv:2603.09972.

Nicholas Shea. Exploitable isomorphism and structural representation. *Proceedings of the Aristotelian Society*, 114:123–144, 2014. doi: 10.1111/j.1467-9264.2014.00367.x.

Alex F. Spies, William Edwards, Michael I. Ivanitskiy, et al. Transformers use causal world models in maze-solving tasks. *arXiv preprint arXiv:2412.11867*, 2025.

Melody Stevinson, Lucas Prieto, Melih Barsbey, and Tolga Birdal. Adversarial attacks leverage interference between features in superposition. *arXiv preprint arXiv:2510.11709*, 2025.

Matthieu Tehenan, Christian Bolivar Moya, Tenghai Long, and Guang Lin. Linear spatial world models emerge in large language models. *arXiv preprint arXiv:2506.02996*, 2025. doi: 10.48550/arXiv.2506.02996. URL https://arxiv.org/abs/2506.02996.

Jayden Teoh, Manan Tomar, Kwangjun Ahn, Edward S. Hu, Tim Pearce, Pratyusha Sharma, Akshay Krishnamurthy, Riashat Islam, Alex Lamb, and John Langford. Next-latent prediction transformers learn compact world models, 2026. URL https://arxiv.org/abs/2511.05963.

Keyon Vafa, Justin Y. Chen, Jon Kleinberg, Sendhil Mullainathan, and Ashesh Rambachan. Evaluating the world model implicit in a generative model. *arXiv preprint arXiv:2406.03689*, 2024.

Keyon Vafa, Peter G. Chang, Ashesh Rambachan, and Sendhil Mullainathan. What has a foundation model found? using inductive bias to probe for world models. In *International Conference on Machine Learning*, 2025. arXiv:2507.06952.

Iwan Williams. Can structural correspondences ground real-world representational content in large language models? *Mind & Language*, pp. 1–19, 2026. doi: 10.1111/mila.70018.

Daniel Wurgaft, Can Rager, Matthew Kowal, Vasudev Shyam, Sheridan Feucht, Usha Bhalla, Tal Haklay, Eric Bigelow, Raphael Sarfati, Thomas McGrath, Owen Lewis, Jack Merullo, Noah Goodman, Thomas Fel, Atticus Geiger, and Ekdeep Singh Lubana. Manifold steering reveals the shared geometry of neural network representation and behavior. *arXiv preprint arXiv:2605.05115*, 2026. doi: 10.48550/arXiv.2605.05115.

Sirui Xia, Aili Chen, Xintao Wang, Tinghui Zhu, Yikai Zhang, Jiangjie Chen, and Yanghua Xiao. Can LLMs learn to map the world from local descriptions? In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2823–2845. Association for Computational Linguistics, 2026. URL https://aclanthology.org/2026.acl-long.128/.

Yifan Zhang, Wenyu Du, Dongming Jin, Jie Fu, and Zhi Jin. Finite state automata inside transformers with chain-of-thought: A mechanistic study on state tracking. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13603–13621. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.668. URL https://aclanthology.org/2025.acl-long.668/.

A Intersection features

A.1 Decodability

Intersection representations identify the current position. We read the current intersection two ways: by its nearest centroid and with a linear softmax probe. Both use the same training states

per intersection and are tested on held-out rides. An intersection's centroid c_v is its mean residual; subtracting the global mean gives its diff-means direction, $u_v = c_v - \bar{c}$. Across 4,000 rides (201,779 states), nearest-centroid accuracy peaks at 0.997 at layer 18, versus 0.981 for the probe. The probe remains accurate deeper in the network while nearest-centroid decoding declines (Fig. 8a). In a separate 6,000-ride evaluation, 99.6% of the 4,497 observed intersections decode at accuracy ≥ 0.9 (Table 3); the lowest accuracy is 0.778. We use layer 18 to read position.

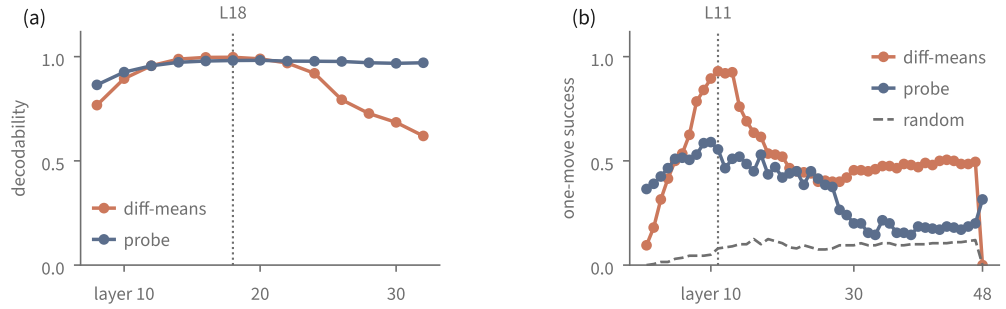


Fig. 8: **Reading and steering position across layers.** (a) Held-out top-1 decode accuracy. (b) One-move success after minimal teleportation on 200 scenes, with $\alpha = \beta = 1.5$; *end* competes with the eight moves. Dotted guides mark the selected reading layer (18) and steering layer (11). The dashed curve is the random-node edit control for diff-means steering.

A.2 Causal Interventions

The minimal teleportation test. We test whether editing the model's position changes its next move as if it were at the new intersection. We choose two intersections X and T one move from the same goal D, each requiring a different move to reach it (Fig. 9). A prompt $[O, D, m_1]$ takes the taxi from origin O to X. At the m_1 token, we subtract the position feature of X and add that of T: $h \leftarrow h + \alpha u_T - \beta u_X$, once at layer 11, which gives the highest teleportation success in the layer sweep (Fig. 8b). We then take the highest-scoring token among the eight moves and *end*. The test succeeds if this move legally reaches D from T. We test both diff-means and probe directions.

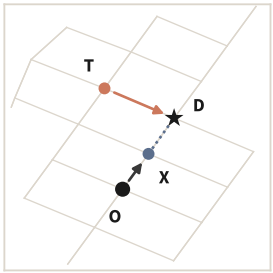


Fig. 9: **Minimal teleportation test.** After the taxi moves from O to X, we edit its position toward T. Success means its next greedy prediction is the orange move from T to the goal D, rather than the dotted blue move from X.

Table 2: **Teleportation success rate on 200 scenes at layer 11.** Rows vary the strength α of the added feature u_T; columns vary the strength β of the subtracted feature u_X. Bold marks the chosen pair.

$\alpha \backslash \beta$	1	1.5	2	2.5	3
1	0.765	0.825	0.690	0.595	0.490
1.5	0.885	**0.930**	0.895	0.750	0.650
2	0.885	0.910	0.900	0.860	0.755
2.5	0.845	0.875	0.890	0.870	0.835
3	0.775	0.850	0.870	0.865	0.830

Most intersection features causally guide the next move. For 99.3% of the 4,202 testable intersections, injecting its feature makes the model choose the move from that intersection to the goal in at least one scene; 94.0% succeed in more than half their scenes (Table 3). This evaluation covers all 24,877 eligible scenes, using diff-means directions with $\alpha = \beta = 1.5$. The correct move's probability also increases in 97.9% of scenes, rising from 10.0% to 56.9% on average over the same

eight moves plus *end*. On the 200 scenes used to compare edit strengths, the model chooses the move from T to D in 93.0% of cases, against 0.0% without an edit and 8.0% with random intersection features. Success remains high around strengths 1.5–2 (Table 2). Probe directions, scaled to the residual norm, reach at most 59.0% across their tested layers (Fig. 8b).

Table 3: **Most intersection features are causal.** Teleportation succeeds for 99.3% of the $4{,}202$ testable intersections on at least one scene, and for 94.0% on more than half. Decoding accuracy is reported below for the $4{,}497$ intersections observed in held-out rides. Rows within each group overlap.

criterion	intersections	share
teleport succeeds on any scene (L11)	4,172	99.3%
teleport succeeds on $> 50\%$ of scenes	3,951	94.0%
teleport succeeds on every scene	2,764	65.8%
decode accuracy ≥ 0.9 (L18)	4,478	99.6%
decode accuracy $= 1$	4,026	89.5%

A.3 Spatial structure

Intersection features encode geographic location. A linear readout predicts latitude and longitude for held-out intersections, reaching $R^2 = 0.988$ for both coordinates at layer 6 (Table 4). We fit ridge regression on 80% of mapped intersections and test on the remaining 20%, using mean-centered feature directions and standardized coordinates (penalty 100, seed 0). Fig. 10 shows the predicted coordinates at layer 6. The same readout is much less accurate on a randomly initialized model of the same architecture. We do not show causal exploitation of this encoded spatial structure here; evidence of causal use of spatial information comes from the goal compass (Section 2.2; App. C.2).

Table 4: **Coordinates are linearly readable from intersection features across layers.** Latitude/longitude R^2 on the 20% of intersections held out from fitting. The random-init control uses layer 6.

layer	6	8	10	12	14	16	18	20	random-init
latitude	**.988**	.984	.977	.971	.968	.964	.961	.960	.534
longitude	**.988**	.985	.979	.975	.972	.969	.966	.965	.474

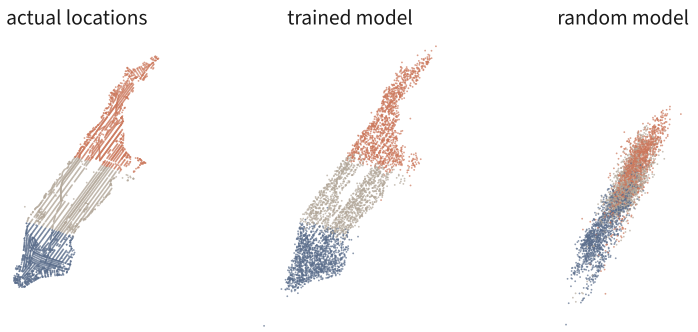


Fig. 10: **A linear readout of intersection features recovers Manhattan's spatial layout.** Actual locations (left) and coordinates predicted from layer 6 features in the trained (middle) and random (right) models, on the same scale. Colors mark three geographic bands along Manhattan's long axis, kept fixed across panels. Held-out latitude/longitude R^2 is $0.988/0.988$ for the trained model and $0.534/0.474$ for the random model. Maps include intersections used to fit the readout.

A.4 SUPERPOSITION AND AFFORDANCE PACKING

Intersection features occupy a small subspace and overlap. At layer 18, 244 dimensions capture 90% of the variance among 4,516 intersection directions. Their mean absolute cosine similarity is .073, compared with .020 for random directions in the same 1,600-dimensional space (Table 5). We use unit-normalized diff-means directions and measure dimensionality from their covariance eigenvalues λ_i; the participation ratio is $(\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$.

Intersection features are packed by affordance. Feature clusters align more closely with legal-move sets (the intersections' affordances) than with geographic location. Fig. 11 shows examples of superposed features at different locations that share the same legal moves. We compare feature clusters with affordance labels and geographic clusters using adjusted mutual information (AMI), with 104-cluster K-means for both features and coordinates. We also compare each feature with normalized affordance-group means, which include the feature being scored. For within-affordance cosine, we first average within each group of at least five members, then average equally across groups.

Superposition packing also favors neighboring intersections. We sample pairs of intersections with the same legal moves, using groups with at least ten mapped intersections. The geographically closest fifth of these pairs has mean feature cosine similarity .290, compared with .213 for the farthest fifth (Table 5).

Table 5: **Intersection features overlap and group by legal moves.** Measurements use layer 18 features. Affordance means the set of legal moves. Map-near and map-far pairs are the closest and farthest fifths of the pooled sample of same-affordance pairs, ranked by geographic distance.

quantity	value
dimensions holding 90% of the variance	244
participation ratio	101
mean pairwise $\lvert\cos\rvert$, features / random	.073 / .020
mean nearest-neighbor cosine	.668
mean within-affordance cosine	.234
nearest affordance mean is the feature's own	91.4%
nearest feature shares affordance	61.0%
clustering AMI, affordance / space	.574 / .192
same-affordance feature-neighbor is closer than a random one	85.8%
within-affordance cosine, map-near / map-far pairs	.290 / .213

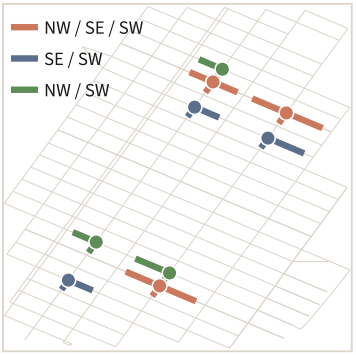


Fig. 11: **Superposed features at intersections with the same legal moves.** Each color marks three such intersections. Within each group, two feature directions have cosine similarity above 0.75 to the third. Colored segments show their legal outgoing moves.

B STREET CONNECTIVITY

B.1 LEGAL MOVES

Intersection features encode which moves are legal. Using the logit lens, we classify moves with probability above 1% as legal. This recovers the complete legal-move set for 99.4% of intersections at layer 31. We apply the model's final layer normalization and output projection to each intersection centroid c_n, then normalize over the eight moves. Fig. 12 illustrates the separation between legal and illegal move logits.

Legal moves are also readable from intersection features at earlier layers. A linear probe trained on 80% of intersections achieves F1 = 0.989 on the held-out 20% at layer 18 (Table 6). We use ridge regression with a fixed split (seed 0).

Table 6: **Legal-move F1 by layer.** The linear probe is fitted on training intersections and scored on held-out intersections; logit-lens uses the model's output projection. Each entry uses the threshold maximizing F1 on its reported scores. These scores pool intersection–move pairs, rather than requiring the complete legal-move set to be correct.

layer	18	20	24	28	32	36	44
linear probe	**.989**	.997	.998	.999	.999	.999	.999
logit-lens	.937	.962	.993	.999	.999	.999	.999

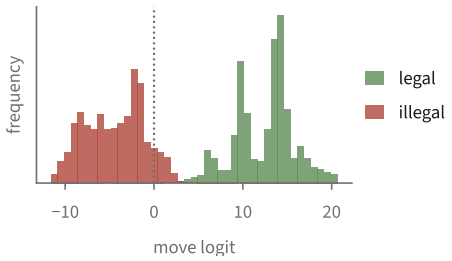


Fig. 12: **Move logits from centered intersection directions at layer 32, split by legality.** We apply the logit lens to $c_n - \bar{c}$. Legal moves generally receive positive logits and illegal moves negative logits.

B.2 Where the streets lead

Street steering test. We inject an intersection feature, feed a move, and measure which intersection feature grows most afterward (Fig. 13a). The other prompt states are averaged to avoid tying the test to a particular ride. Across $8{,}934$ tested streets, the correct next intersection ranks first in 76.6% of cases and among the top five in 93.3% (Fig. 14). The four-token prompt contains an origin, destination, position slot, and tested move. We replace the first three states with their layer-specific means from held-out rides, then add the source intersection's normalized diff-means direction at the position slot at every layer from 1 through 18. Its strength is α times the median direction norm, with $\alpha \in \{0, 1, 2, 4\}$. At the move token, we rank all $4{,}516$ normalized intersection directions by the increase in their layer-18 readout score from $\alpha = 0$ to $\alpha = 4$. We also check whether the correct next intersection's score increases monotonically across the four strengths. The tested streets come from $3{,}820$ intersections with at least two tokenized exits and available source and next-intersection features.

Street probing test. We train eight linear probes, one per move, to read ordinary ride states and predict the feature of the intersection that move would reach (Fig. 13b). We match each prediction to the closest of the $4{,}516$ intersection directions by cosine similarity. At layer 15, accuracy is 89.2% on queries from held-out source intersections, including moves not taken during the ride. We hold out 25% of source intersections, fit on $4{,}000$ training rides, and evaluate on $4{,}000$ held-out rides. The held-out sources' states are excluded from fitting; their features remain among the candidate answers. Each probe maps $h - \bar{c}$ to a next-intersection direction $c_v - \bar{c}$: we initialize it with ridge regression, then fit with cross-entropy over cosine scores. We compute accuracy over $113{,}998$ state–move queries covering $2{,}412$ distinct streets. The same protocol predicting the current intersection reaches 99.3%; shuffled targets give 0.03%.

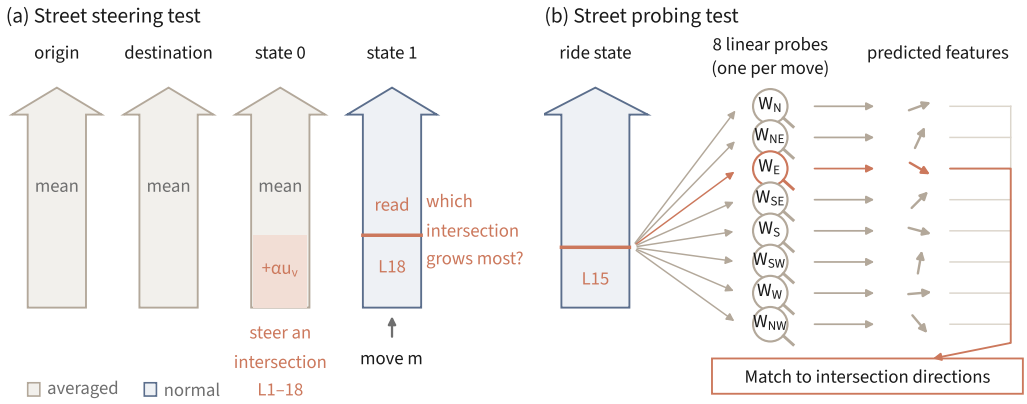


Fig. 13: **Two tests of street connectivity.** (a) The steering test asks whether an intersection feature and a move activate the feature of the correct next intersection. We inject an intersection feature into the averaged state 0 and feed move m, then measure which intersection feature grows most at state 1. The other prompt states are averaged (beige) to avoid tying the test to a particular ride; state 1 evolves normally (blue). (b) The probing test asks whether street connections can be read from the model's state during an ordinary ride. Eight linear probes, one per move, predict the feature of the intersection that move would reach. We match each output to the closest intersection feature and test on held-out source intersections.



Fig. 14: **Street steering test.** Each tested move is drawn from its source intersection to the street midpoint, so opposite directions can be colored separately. Color shows the correct next intersection's rank among all candidates; the right panel enlarges the boxed area.

C LOCALIZATION AND NAVIGATION

C.1 THE LOOK-BACK WINDOW OVER PAST POSITIONS

The model uses recent position features to locate itself. Removing position information from past states reduces current-position decoding accuracy from 99.8% to 14.8% on 27–34-move rides. Preserving the most recent past position gives 60.0% accuracy, the last six give 91.7%, and the last twenty give 99.2%. We replace each past move token's position-subspace component with that of its move centroid to recompute its cached keys and values. The past forward pass stays fixed: we do not propagate the edit through past tokens, and rerun the model only for the final token. The subspace contains the top singular directions of the diff-means matrix accounting for 90% of its squared singular values. We preserve the orthogonal component, the origin and destination states, and the final token, and decode at layer 18. The same edit in a random subspace of matched dimension leaves accuracy at 99.8%; subtracting past move diff-means directions leaves 99.0%. We leave the final move token unchanged, so the model can still combine that move with a past position.

Past position features matter more on longer rides. We repeat the ablation at different ride lengths, removing all past position features. Current-position decoding accuracy falls from 69.2% at 2–4 moves to 14.2% at 43–60 moves (Fig. 15).

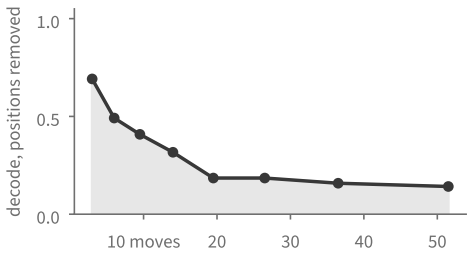


Fig. 15: **Without past positions, decoding worsens as the ride gets longer.** We ablate past position features while preserving the origin, destination and final token, then decode the current intersection at layer 18.

C.2 THE GOAL COMPASS

The goal compass encodes the direction to the goal. We fit a two-dimensional plane on training rides and use the angle within it to predict goal bearing on held-out rides. The median decoding error is $18.1°$ at layer 16 (Fig. 16a). To fit the plane, we center training states within each intersection to reduce the contribution of intersection identity, average them in 16 bearing bins, and combine the centered bin means μ_b, with bin-center bearings θ_b, into two axes:

$$v_{\cos} = \sum_{b=1}^{16} \cos(\theta_b)\mu_b, \qquad v_{\sin} = \sum_{b=1}^{16} \sin(\theta_b)\mu_b.$$

We normalize each axis to unit length and decode bearing as $\operatorname{atan2}(h^\top v_{\sin}, h^\top v_{\cos})$, where h is the residual minus the fixed mean stored during fitting; evaluation does not use the current intersection's identity. We fit on $6{,}000$ training rides and evaluate on $3{,}000$ held-out rides, excluding states fewer than three moves from the goal. Bearings use geographic coordinates with longitude corrected for latitude. (Exploratory analysis with a sparse autoencoder revealed features sensitive to goal bearing, motivating our extraction of the compass using differences of means.)

The model follows the steered direction. We set the compass to a target bearing at each move, allow up to 12 greedy legal moves, and measure the direction from the ride's start to its endpoint. Across 16 target bearings, the median error is $18.1°$ and the mean is $24.5°$. Steering works across the tested bearings, with the highest alignment near Manhattan's long axis (Fig. 16b). The edit removes the existing compass component with strength β and adds the target direction with strength α, scaled by the remaining residual norm. We orthonormalize the two axes before projecting out the compass

component. We use $\alpha = \beta = 1$ at layer 15; Table 7 compares strengths. We retain 1,460 rides with at least two moves and nonzero displacement, and compute the mean and median of their individual angular errors. As a graph-only reference, the best endpoint reachable in exactly 12 legal moves has a median error of 1.0°. The north/south examples in Fig. 2b use a stronger temporary edit at layer 18 ($\alpha = 16$, $\beta = 1$): after 3 initial moves, we steer for 12 moves and release the edit for up to 30 further moves. Each phase starts a fresh prompt at the current intersection with the same goal.

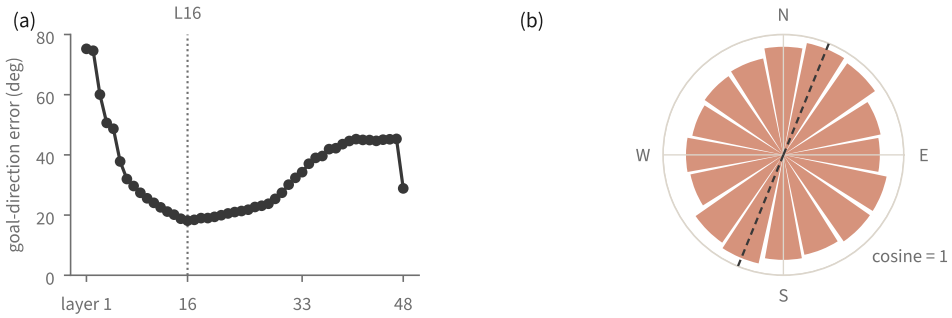


Fig. 16: **The goal compass encodes goal direction and guides travel.** (a) Error in the goal bearing decoded from held-out ride states, by layer. (b) The model follows the imposed direction across the 16 tested bearings. Each wedge points in a direction imposed on the compass. Its radius shows how closely the model travels in that direction: mean cosine alignment, with 1 for perfect alignment. Alignment is highest near Manhattan's long axis (dashed), estimated from the map coordinates.

Table 7: **Choosing the compass edit strength.** At layer 15, β scales removal of the existing compass component and α scales the added target direction. Entries give mean $\cos(\text{heading} - \text{target})$: 1 means aligned, 0 perpendicular, and -1 opposite. We use 60 start–goal pairs and target bearings 45° and 225°; the 16-bearing sweep uses 100 pairs. Bold marks the chosen setting.

	β=0	β=0.5	β=1	β=1.5	β=2
α=0.25	.701	.796	.834	.847	.845
α=0.5	.904	.913	.892	.899	.894
α=1	.905	.919	**.920**	.922	.917
α=1.5	.843	.846	.857	.857	.826
α=2	.505	.513	.512	.484	.434

Intersection features favor legal moves; the compass favors goalward moves. We read move preferences from intersection features using the logit lens, then compare the effect of compass edits pointing toward versus away from the goal. Across 2,817 origin–goal prompts, centered intersection features at layer 32 give logits near +12 for legal moves and −4 for illegal moves. For the compass, we set its layer-16 component toward the goal, then away, and take the difference in output logits. Among legal moves, the goalward-minus-awayward contrast is 0.11 for the intersection feature and 1.94 for the compass intervention. The compass accounts for 95% of the sum of these two measured contrasts. We define goalward moves as those within 60° of the goal bearing and awayward moves as those beyond 120°, using each move's average geographic direction and excluding the middle band. This also assigns a direction to illegal moves. Compass edits use $\alpha = \beta = 1$: the added direction has the residual's norm after compass removal.

Removing the compass impairs goal-reaching but preserves legal moves. We remove the compass plane at layer 18 and compare with removing a random plane of the same dimension. For the immediate next move, compass removal changes the highest-scoring legal move in 33.0% of 2,500 origin–goal prompts (random: 0.2%), while the highest-scoring move remains legal in all three conditions. For complete rides, goal-reaching falls from 94% to 23% in the farthest starting-distance bin (18–32 moves), while 99.5–99.7% of moves remain legal across the distance bins (Table 8). The random-plane control remains close to baseline. We sample from the full vocabulary at temperature 1

for up to 128 moves and count a ride as successful if it visits the goal. Fig. 17 shows four examples that fail after compass removal.

Table 8: **Compass removal reduces goal-reaching while moves remain legal.** Starting distance is the minimum number of legal moves to the goal. The four distance bins contain 31, 50, 115, and 466 tasks. Removing a random plane of the same dimension serves as a control.

start distance (moves)	3–6	7–11	12–17	18–32
reaches goal:				
with compass	1.00	1.00	0.99	0.94
compass ablated	0.68	0.54	0.38	**0.23**
random-plane ablated	1.00	1.00	0.98	0.94
moves that are legal:				
compass ablated	0.995	0.997	0.996	0.995



Fig. 17: **Without the compass, the model follows streets but misses the goal.** Four selected rides after compass removal; circles mark origins and stars mark goals. The corresponding baseline rides reach their goals.

D Other mechanisms

D.1 The at-goal feature

The at-goal feature fires when the model is at the destination. We compare the same ride state with two goals: its current intersection and a node at least 15 moves away. This holds the origin and move prefix fixed while changing whether the ride is at its goal. We sample 1,500 states from training rides. We use half of these pairs to fit a difference-of-means direction and a midpoint threshold, then test the readout on the other half. Accuracy reaches 100% at layer 20, tied with several later layers (Fig. 18a). The model's mean $P(\text{end})$ is 0.55 at the goal and below 0.001 away.

Adding the direction makes the model stop, even away from the destination. At 400 evaluation states with a distant goal, we add the layer-20 direction at strength $\alpha\|h\|$. The stop token becomes the highest-scoring token in 98.3% of states at $\alpha = 1$ and 100% at $\alpha = 2$, versus 0% without the edit. A random direction with the same injection norm gives a 0% stop rate across the tested strengths (Fig. 18b).

The at-goal feature is distinct from a general stopping signal. We sample 3,000 rides from origins 28–45 moves from the goal and apply the fitted layer-20 threshold at their final states. The feature fires at 93% of successful stops, but at 13% of wrong-stop or generation-limit outcomes, despite high $P(\text{end})$ in both groups (Table 9). The latter group has median length 99 moves. Illegal-move outcomes are excluded; rides without a stop are capped at 128 moves.

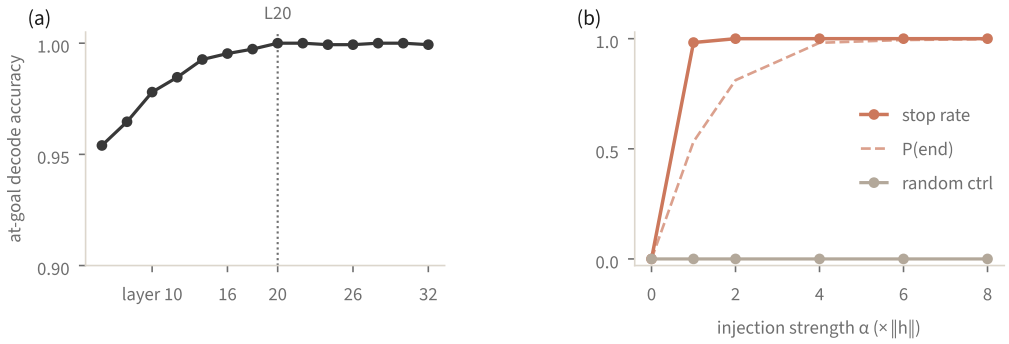


Fig. 18: **The at-goal feature detects arrival and can trigger stopping.** (a) Held-out accuracy when the goal is set to the current intersection or a distant node. (b) Stop-token argmax rate and mean probability after injection at states away from the goal; the random control shows its argmax rate.

Table 9: **At-goal readout at final states of self-generated rides.** Generation-limit outcomes reach the 128-move cap without a stop token or illegal move.

outcome	feature fires	$P(\text{end})$	n
stops at the goal	0.93	0.80	2411
wrong stop or generation limit	0.13	0.89	383

D.2 THE COMMIT-TO-GOAL FEATURE

A single direction controls whether the model stops at the goal. From training rides, we collect states that end at the goal and states that pass through it within the first 40% of the ride, with at least five moves remaining. A difference-of-means direction fitted on half of each class distinguishes the remaining states with 83.6% accuracy at layer 16. We orient it toward continuing: adding it promotes exploration, while subtracting it promotes commitment to the goal. Adding it at 300 evaluation states where the training ride ends reduces mean $P(\text{end})$ from 0.62 to 0.002 at $\alpha = 2$, where the injection norm is $\alpha\|h\|$. The random control gives 0.52 at that strength (Fig. 19a).

The same direction also controls how much the model meanders. We test whether the same edit changes the route before arrival, adding it at each move during full-vocabulary, temperature-1 generation on 120 origin–goal pairs 10–25 moves apart, with a 128-move cap. At $\alpha = -0.9$, the mean path length among rides ending at the goal falls from 3.72 to 1.26 times the shortest path, while the fraction ending there changes from 94.7% to 88.6%. At $\alpha = 0.5$, these values are 4.64 and 41.7%. We average over three generation seeds and compare with three norm-matched random directions (Fig. 19b). Fig. 20 illustrates the route changes for one selected pair.

Commitment to the goal increases on longer stress rides. For this readout, we fit the layer-16 direction on all collected stop and continue states, then project states from $6{,}000$ self-generated stress rides onto it, with the sign reversed so higher means more committed (Fig. 21). The projection increases late in rides. This is consistent with the hypothesis that, as the model approaches the roughly 100-move training horizon, it commits more strongly to reaching the goal within the remaining moves.

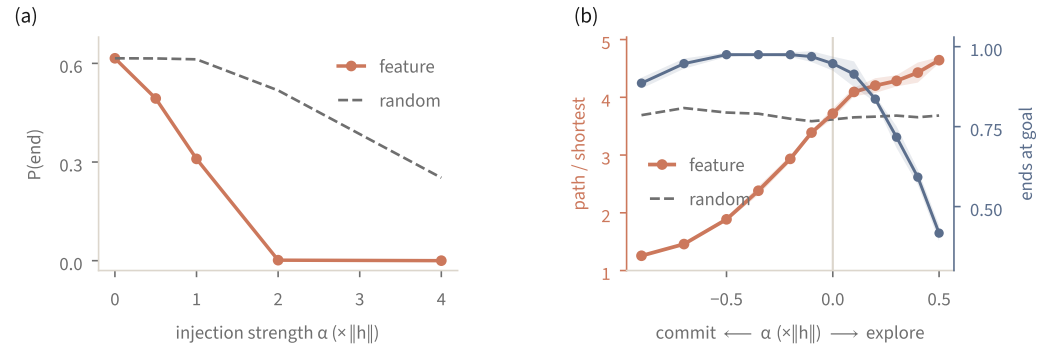


Fig. 19: **One direction controls stopping and how much the model meanders.** (a) Stop probability after adding the continue direction at terminal training-ride states. (b) Path length relative to the shortest path (clay), conditional on ending at the goal, and the fraction ending there (slate). Negative strengths favor commitment; positive strengths favor exploration. Bands show one standard deviation across three generation seeds; dashed lines show random-direction controls.

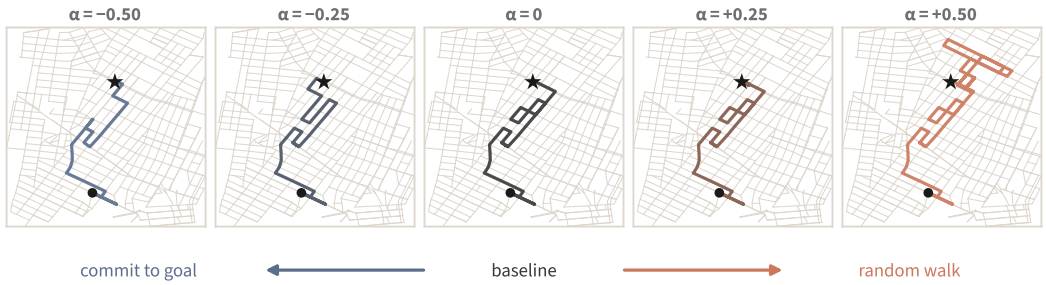


Fig. 20: **Steering changes the route on a fixed origin–goal pair.** One selected origin–goal pair at five steering strengths, with a 200-move cap. Circles mark the origin and stars the goal. Negative strength shortens this route; positive strength adds loops.

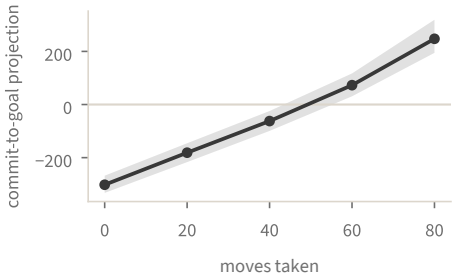


Fig. 21: **Commit-to-goal projection by depth on stress rides.** The line shows the median and the band the interquartile range; higher values point toward committing to the goal.

E THE STRESS TEST

E.1 THE STRESS TEST IS OUT OF DISTRIBUTION

Stress rides reach the deep-and-far regime. Among reachable pairs in the 6,400 released rides, median origin–goal distance is 32 moves, versus 9 in 50,000 held-out training-distribution rides (Fig. 22a). The model therefore often remains far from the goal late in a stress ride. The deep-and-far region, with at least 60 moves taken and 20 moves remaining, contains 5.4% of stress states but

only 0.1% of held-out states (Fig. 22b). For this comparison, we use 8,000 held-out rides and 20,000 generated stress rides, sampled from the full vocabulary at temperature 1 with a 128-move cap. We sample pairs from the released pool to approximately match its distance histogram, treating reversed pairs as duplicates.

Failures concentrate in the deep-and-far regime. In the released rides, failure increases with origin–goal distance (Fig. 22a). In our generated stress rides, the deep-and-far region accounts for only 5.4% of all visited states but 57.4% of off-graph outcomes. Failures thus occur disproportionately in a regime almost absent from training.

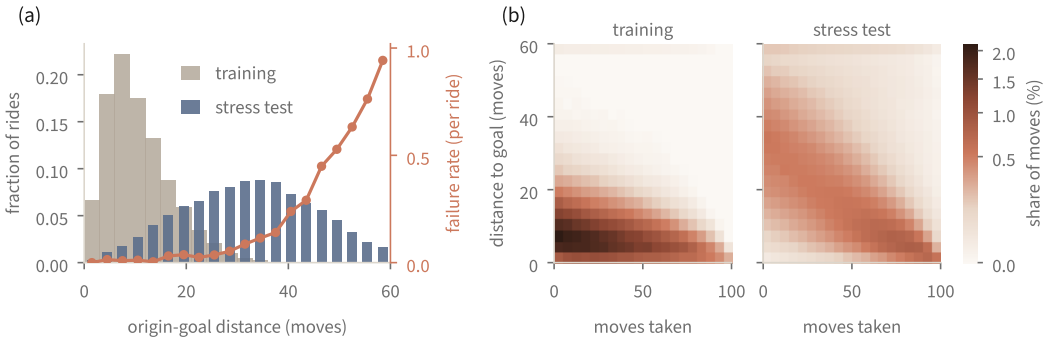


Fig. 22: **The stress test shifts both pair distances and visited states.** (a) Origin–goal distances in the released stress reference and held-out training-distribution rides, with reference failure rate on the right axis. (b) Visited-state occupancy for held-out rides and our generated stress set. Heatmaps show depths up to 100; their top distance bin includes all distances of 57 moves or more.

E.2 Write strength and noise

Key information for Fig. 6. We measure 12,000 stress rides at layer 18. After subtracting the global mean, the write is the projection onto the unit true-intersection direction; noise is the root-mean-square projection onto 96 fixed sampled intersection directions. Crowding counts other directions with cosine similarity above 0.5 to the true feature. Route surprise is the mean negative log-probability of the moves taken so far, normalizing over the eight move tokens. For the goal-swap experiment, we keep the origin and moves fixed in 800 stress prefixes and replace only the destination, choosing one 1–5 or 30–69 moves from the current intersection. Moving the goal from near to far weakens the layer 18 write in 86.5% of cases, with a median paired decrease of 59.4. The substantial change from this single-token swap suggests a learned adjustment of the position write, whose purpose remains unclear. The swap also changes route probability, so it does not isolate distance from other effects of goal conditioning.

Why we interpret the remainder as noise. We test whether activity beyond the true-position write is concentrated on nearby intersections or forms a stable part of the feature. After subtracting the write, absolute projections toward intersections one to three moves away are only 25% and 15% larger than toward the rest of the map in 500 clean and 500 illegal states. Comparisons across angles to the true feature, with random-vector controls, also show broadly distributed activity with a modest preference for aligned directions. For stability, we average the whole residual after subtracting the write across visits to each intersection in 1,500 held-out rides. Averaging 24 visits reduces its norm to 30% of a single visit, versus 21% for norm-matched isotropic noise. The estimated fixed component accounts for 4.4% of its energy (median over intersections with at least 12 visits, corrected for finite sampling), including feature-estimation error. The remainder thus varies substantially across visits and is not confined to nearby intersections, although it retains some structure.

E.3 Illegal moves and recovery

Additional details for the main-text experiments. Table 10 gives the ordered classification rule. Cases with total illegal-move probability below 0.001 are reported separately as low-mass unlucky

draws; all cases remain in the denominator. All three rows of Fig. 5 use the same failure cases: rides with at least seven preceding moves, measured immediately before the sampled illegal move, which need not be the highest-scoring move. We apply no further sampling or illegal-probability cutoff for these rows. Clean and recovering references are measured at their respective time-zero states. For the layer-18 repairs, we use full-vocabulary probabilities and report $1 - \sum P^{\text{after}}_{\text{illegal}} / \sum P^{\text{before}}_{\text{illegal}}$. The wrong-node edit removes the strongest non-true direction after orthogonalizing it against the true unit direction u_T. With $v = h - \bar{\mu}$ and P projecting onto the top 244 singular directions of centered centroids, noise clearing subtracts $Pv - \frac{(Pv)^\top u_T}{(Pu_T)^\top u_T} Pu_T$, preserving the true write and activity outside the subspace. Write restoration raises the true-position write to 509 if it is below that value. Controls use matched edit norms along random position-subspace directions orthogonal to u_T. In Fig. 3b, we cap the write at 300, 240, 200, or 150 (plus an uncapped control), then add Gaussian noise with coordinate standard deviation 0, 30, 67, or 120 in that subspace. We add noise after capping the write, so the noise can also change the true-position write. We record the highest-scoring move among the eight move tokens and the intersection with the strongest activation.

Individual stress rides illustrate the failure categories. Fig. 23 shows two trips per category. Both galleries select distinct origin–goal pairs near the category medians of final-state write, noise and wrong-node cosine, adding stopping activation for give-up slips. We minimize the root-mean-square deviation scaled by each quantity's interquartile range, subject to map readability.

Table 10: **Failure categories, applied in the order shown.** Here $v = h - \bar{\mu}$, $u_n = (\mu_n - \bar{\mu})/\|\mu_n - \bar{\mu}\|$, T is the true intersection, and $L = \arg\max_n v \cdot u_n$. The same direction scores define the strongest wrong node in the figures. A supplier is a non-true node among the 13 highest direction scores where the highest-logit illegal move is legal. Shares use all 1,681 cases.

kind	condition	share
low-mass unlucky draw	$P_{\text{illegal}} < 0.001$	6.0%
give-up	else $v \cdot u_{\text{stop}} \geq 38.8$	16.2%
silent slip	else $L = T$ and a supplier exists	23.9%
fatal slip	else $L \neq T$, $u_L \cdot u_T > 0.2$	24.7%
full corruption	else $L \neq T$, $u_L \cdot u_T \leq 0.2$	28.7%
unclassified	else $L = T$ without a supplier	0.5%

Steering toward a wrong intersection changes which move is selected. On 400 clean deep states, we add a co-active wrong-node direction across layers 10, 12, 14, and 16, scaled by each residual norm. The greedy move becomes illegal at the true node in 18%, 42%, and 48% of cases at total strengths 0.5, 1, and 2; in each case it is legal at the targeted node. Steering toward the true node induces no illegal moves. The random-direction control reaches 12% at strength 2.

Illegal moves can be supported by a bag of co-active features. A corrupted position code can activate several wrong-intersection features at once. We call the k most active wrong features a *bag of co-active features*, and ask whether the highest-scoring illegal move is legal at any of their intersections. These bags support the move more often than equally sized bags of random nodes, especially for small k (Fig. 24). For silent slips, the most active wrong feature alone supports the move in 55% of cases, compared with 32% for a random node.

Recovery after a slip relies on the look-back window over past positions. We select 400 recovering superposition slip states where the decoded intersection is at least two moves from the true intersection and returns to the true node on the next move, each paired with a clean control at exactly the same depth and remaining goal distance. At that next move, we retain only the last K past position codes and replace older ones with their move-average components in the cached keys and values, without propagating edits through earlier states. With $K = 3$, accuracy is 43.5% after a slip and 74% on clean controls (Fig. 25). Both reach 100% with the full history because we selected cases with correct decoding at that next move. These percentages therefore do not measure how often slips recover in general.

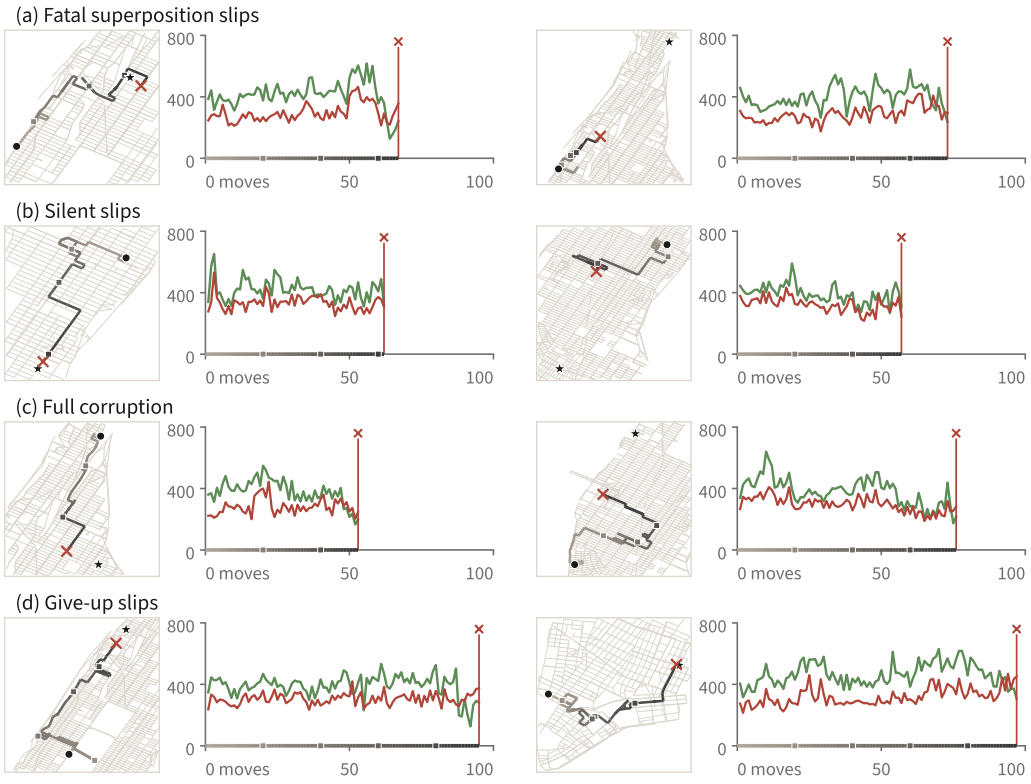


Fig. 23: **Stress trips illustrating four failure modes.** Examples are selected near category medians at the final state. Maps mark the start (black dot), goal (star), and illegal-move attempt (red cross). Matching colored squares mark every 20 moves on the route and time axis. Traces show the layer-18 true-position write (green) and strongest wrong-node activation (red). True- and wrong-node activations often rise and fall together. This is expected under superposition: intersection features have overlapping directions, so strengthening the true-position feature also raises the activation of wrong features that overlap with it.

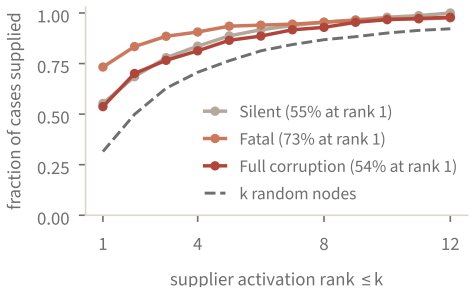


Fig. 24: **The co-active bag supports illegal moves more often than random nodes.** For the bag of k most active wrong-intersection features, curves show how often the highest-scoring illegal move is legal at one of their intersections. The dashed curve uses k random nodes.

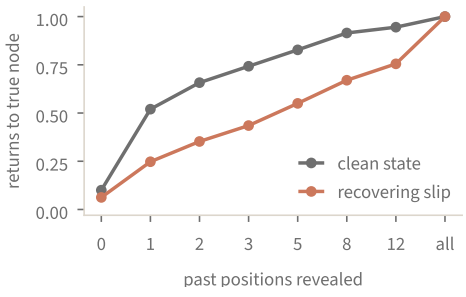


Fig. 25: **Recovering slips need more past positions than clean controls.** Next-state accuracy after removing position information from older moves, on 400 pairs matched exactly on depth and remaining goal distance. Both groups were selected to decode correctly with the full history.

E.4 GIVE-UP SLIPS

The give-up feature promotes stopping. We extract a direction by contrasting states where the model stops away from the goal with late states from successful rides. We take the difference of their

mean residuals at layer 18, remove its position-subspace component, and normalize it. The give-up direction is distinct from the at-goal direction (cosine 0.094 at L18). To test whether this direction promotes stopping, we add it to states from successful rides and measure whether *end* becomes the highest-scoring token. Adding the direction reliably induces stopping, unlike a matched random direction (Fig. 26a). We use 1,164 states sampled every four moves from 60 successful rides, adding $\alpha/4$ times the residual norm at each of layers 10, 12, 14, and 16.

Give-up slips have enlarged residuals and unusually high position noise. The stopping signal can be active while the model still produces an illegal move. In these states, residual norms are larger than in the other failure categories (Fig. 26b). The position code is also noisier, even though the true-position write is stronger than in full corruption. Give-up slips thus combine the stopping signal with a corrupted position code, which can still contain superposed wrong nodes. For classification, we use a threshold of 5% of the median projection onto this direction in states where the model stops away from the goal.

Reducing excess residual activity improves legality. On 32 give-up prefixes, we compare removing the stopping signal with reducing excess activity outside the position and stopping subspaces (Fig. 26c). Removing only the stopping signal reduces stopping probability, but barely changes illegal-move probability. For the residual reduction, we cap the unprotected component at its median norm in healthy states, preserving the position and stopping projections at each edit. This lowers illegal-move probability by 15.2 percentage points beyond matched random edits, while position noise remains high. Both interventions act at every ride-state position across layers 14–47. Local stopping directions are fitted from 32 wrong-stop and 24 healthy states; layer 18 uses the classification direction. We compare with three random directions, matching edit magnitudes at each layer and position.

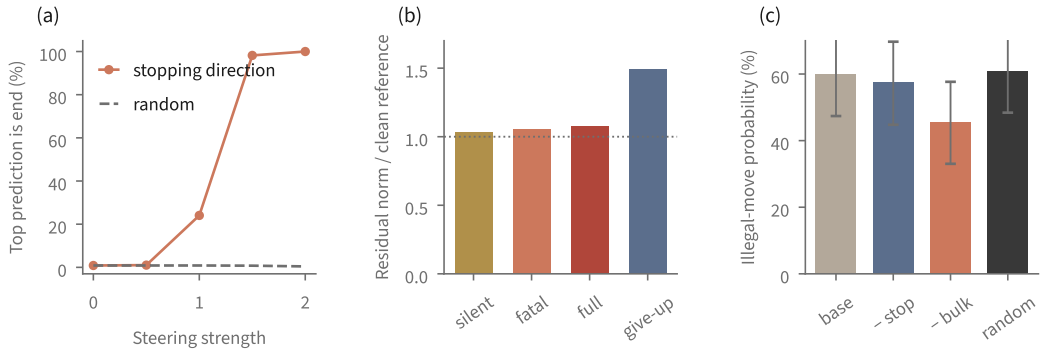


Fig. 26: **Give-up slips combine stopping activity with enlarged residuals.** (a) Adding the give-up direction makes successful-ride states predict *end*. (b) Residual norms across failure categories, relative to clean states. (c) On give-up prefixes, reducing excess residual activity lowers illegal-move probability more than suppressing the stopping signal alone. Random edits match the reduction's magnitude at each layer and position; error bars are 95% bootstrap intervals over prefixes.

F THE DETOUR TEST

Test details. We generate detours using the reference rule: at each non-goal state, with probability 0.75, take the least-likely legal move among those that leave the goal reachable within the remaining budget. Otherwise use the model's greedy prediction. Success requires emitting *end* at the goal. We use 10,000 held-out origin–goal pairs and a 100-move generation cap. The model stops at the goal on 63.6% of rides, makes an illegal move on 25.6%, and stops elsewhere on 10.7%. The main comparison uses the original benchmark implementation; the analyses here use these 10,000 generated rides.

Individual forced moves do not cause a distinct drop in the position write. We compare move-to-move changes in write and noise after imposed and self-chosen moves (Fig. 28a). The distributions

are similar: forced moves do not show a distinct immediate drop in write or rise in noise. This comparison uses the moves as they occur, without matching the two groups. All panels of Fig. 28 exclude states above the stress-fitted give-up threshold.

Repeated forcing creates deep-and-far, unlikely rides with weaker position writes. Goal distance rises to a median of about 30 moves by move 60, then falls as the reachability constraint restricts the available detours (Fig. 27). States with at least 40 moves taken and at least 20 moves remaining account for 32.3% of detour states, compared with 15.0% of stress states and 0.4% of held-out training-distribution states. Within depth bands, the position write is weaker farther from the goal (Fig. 28b). Detour routes also have lower mean move log-probability than stress routes (panel c), and lower route probability accompanies weaker writes (panel d). Repeated forcing thus brings rides into both conditions associated with weak position writes.

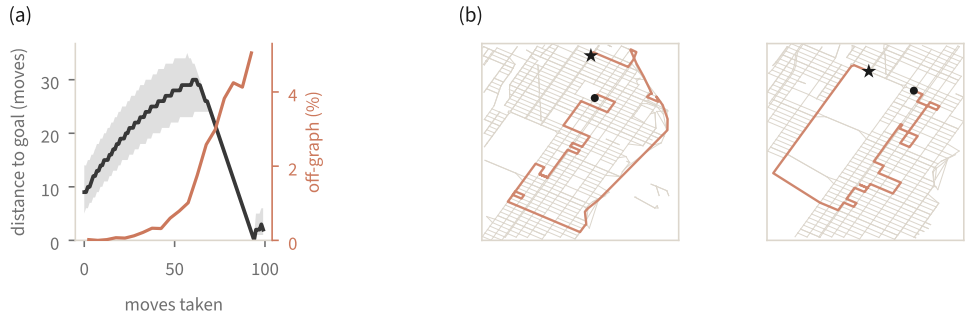


Fig. 27: **Detours first move away from the goal, then return.** (a) Median goal distance over 10,000 rides, with the interquartile range; the clay line shows the percentage of ongoing rides going off-graph in each five-move bin (right axis). (b) Two selected successful trajectories illustrating the reachability-constrained return; dots mark origins and stars goals.

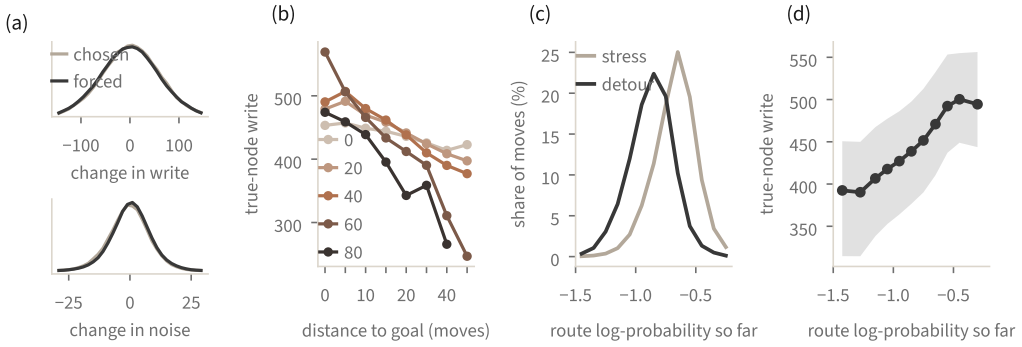


Fig. 28: **Individual forced moves do not immediately disrupt the position code, but repeated forcing creates deep-and-far, unlikely rides on which the position write weakens.** (a) Move-to-move changes in write and noise after imposed versus self-chosen moves. (b) Median write decreases with goal distance within depth bands. (c) Detour routes are less likely than stress routes. (d) Lower route probability accompanies weaker writes; shading shows the interquartile range. Route log-probability is averaged over moves taken so far. States above the give-up threshold are excluded throughout.

The same failure modes appear, with more full corruption. Applying the stress classifier gives 49.0% full corruption, 32.7% fatal slips, 9.0% silent slips, and 9.2% give-up cases (Fig. 29). Clearing position noise while preserving the true-position write, or restoring the write, reduces illegal-move probability in every category. Matched random edits remove at most 9% of illegal probability. Fig. 30 follows two individual detour trips from each category.

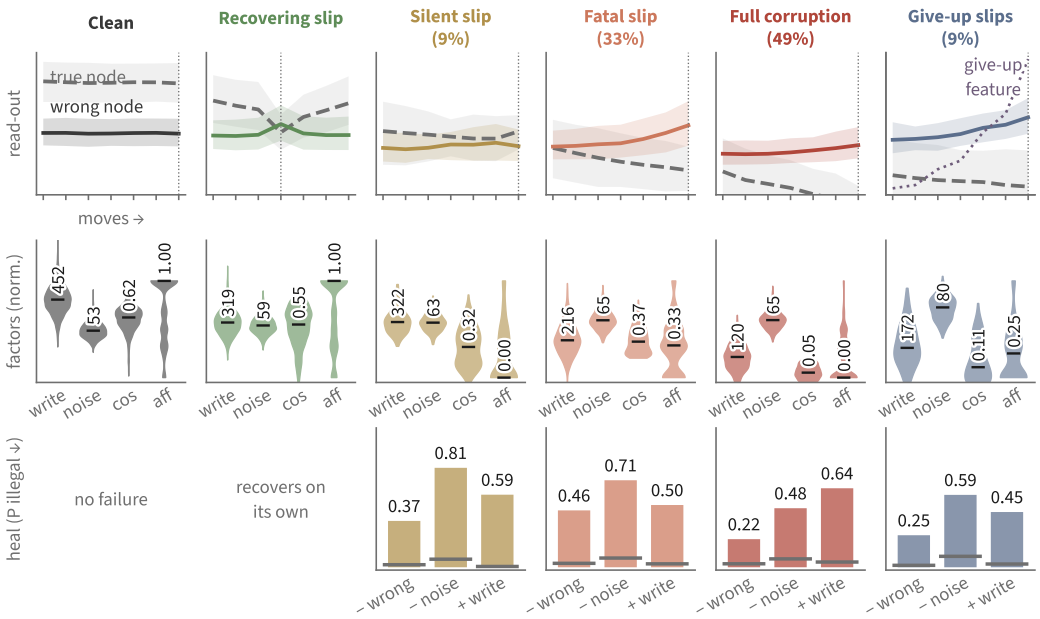


Fig. 29: **Detours exhibit the same failure modes: weak position writes and noise, with recovery under the same position edits.** Columns compare failure categories and reference states. Rows show position readouts around failure or recovery, scaled write/noise/cosine/affordance measurements, and the fraction of illegal probability removed. Gray marks show matched random edits.

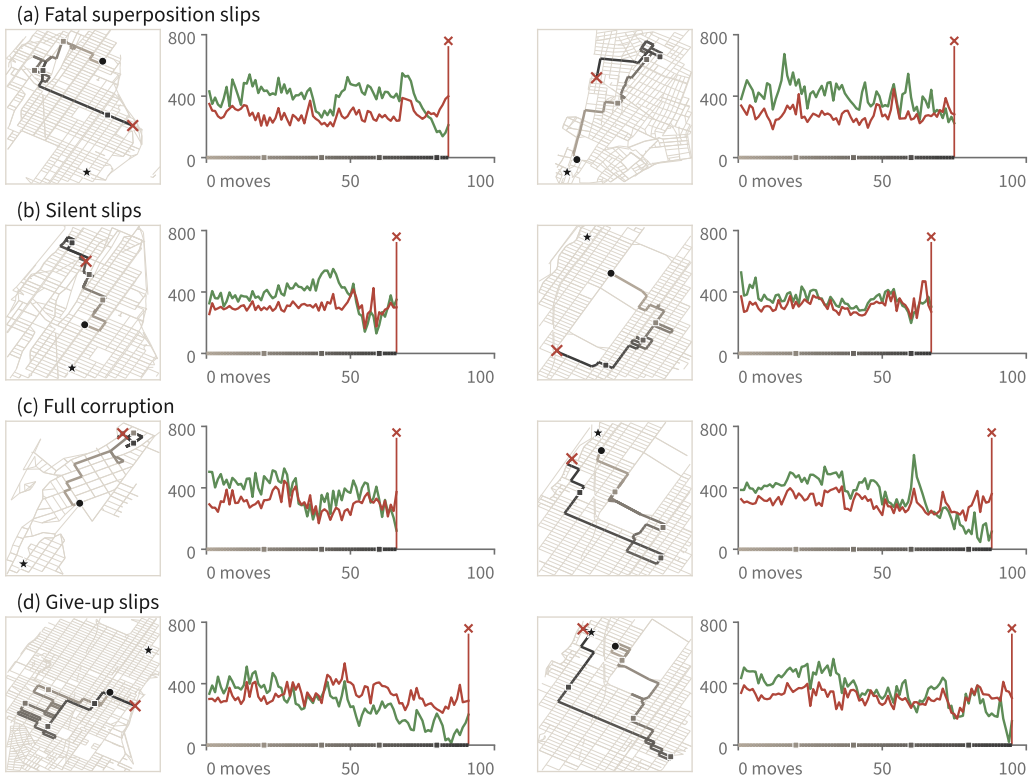


Fig. 30: **Detour trips illustrating four failure modes.** Examples are near category medians at failure. Maps mark the start (dot), goal (star), and illegal-move attempt (cross); squares mark every 20 moves on the route and time axis. Traces show layer-18 true-position write (green) and strongest wrong-node activation (red).

G THE COMPRESSION METRIC

Metric details. Two equal-length prefixes end at the same intersection with the same goal. We sample 30 continuations after the first prefix and score them after the second. A pair passes only if every scored token, including *end*, has probability above $\epsilon = 0.01$ (Fig. 31). The compression score is the fraction of passing pairs. Prefix lengths vary; continuations are sampled at temperature 1 with an epsilon cutoff of 0.01, until stopping or a total sequence length of 128 tokens. The original-code score in the main comparison is 0.524. The analyses below use supplementary runs.

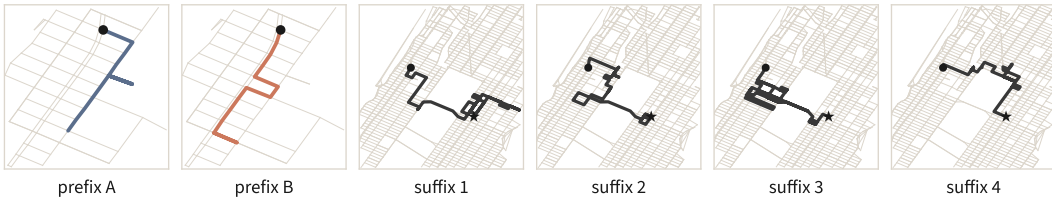


Fig. 31: **Two routes to one intersection, scored through their continuations.** Prefixes A and B have equal length and the same goal. Continuations sampled after A are scored after B; a trial passes only if every token in all 30 continuations clears $\epsilon = 0.01$. The drawing shows a selected subset.

Both prefixes decode correctly even when compression fails. In the 146 sampled-prefix trials inspected here, both prefixes decode to their shared intersection in every pair, including all 94 failing pairs. We read the layer-18 mean-centered residual against normalized intersection directions at the prefix endpoints, before generating the suffix. Table 11 orders the two prefixes by write strength. Failing pairs tend to have more distant goals; correct initial localization does not ensure that the model stays correctly localized throughout the continuation.

Table 11: **Prefix states in passing and failing compression pairs.** The 146 sampled-prefix trials contain 52 passing and 94 failing pairs. Each pair is ordered by write strength. Entries are medians; parentheses give the first and third quartiles for goal distance and prefix length.

at the end of a route	pair passes		pair fails	
	weaker	stronger	weaker	stronger
distance to the goal (moves)	27.5 (21–40)		54 (41–67)	
length of the route (moves)	20.5 (10–34)		18.5 (9–33)	
write on the true intersection	426	457	405	438
noise in the position code	51.3	51.9	51.0	50.9
log-probability of the route	−0.870	−0.832	−0.833	−0.853
a wrong intersection is most active	0%	0%	0%	0%

The continuations reach challenging states and exhibit the same failure modes. The prefix + suffix rides enter the deep-and-far regime, rarely visited in training (Fig. 32b). Within depth bands, the true-position write weakens with distance to the goal (Fig. 32c). About 28% of 21,000 sampled continuations leave the graph. Applying the stress classifier to their failure states gives 21.7% silent slips, 24.9% fatal slips, 39.4% full corruption, and 13.7% give-up cases (Fig. 33). Clearing position noise while preserving the true-position write, or restoring the write, reduces illegal-move probability in every category. Together with the write and noise measurements, these interventions support the same explanation as for stress and detour failures.

Longer suffixes bring more illegal moves and lower compression. To lengthen the suffixes without changing the prefix routes, we move the goal farther away. We keep two 12-move routes fixed at each of 60 intersections with eligible goals in all seven distance bands, changing only the destination token (Table 12, lower block). Scores fall from 0.983 in the nearest band to 0.167 in the farthest; illegal moves rise from 0.06 to 9.18 per thousand checked continuation moves. Both prefixes still decode correctly in $419/420$ pair–goal conditions (99.8%). Across the seven band averages, compression and illegal-move rate are strongly correlated ($r = -0.957$): the longer continuations produce more

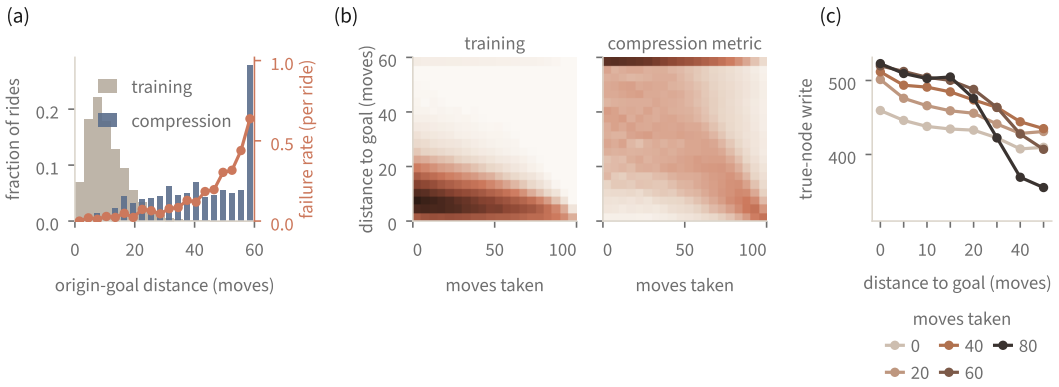


Fig. 32: **Compression generates rides far from the training distribution, where position writes weaken.** (a) Origin–goal distances and off-graph ride rates. (b) Visited-state occupancy by moves taken and remaining goal distance, compared with held-out training-distribution rides. (c) Median true-position write by goal distance, grouped by moves taken in the full prefix + suffix ride; states above the give-up threshold are excluded.

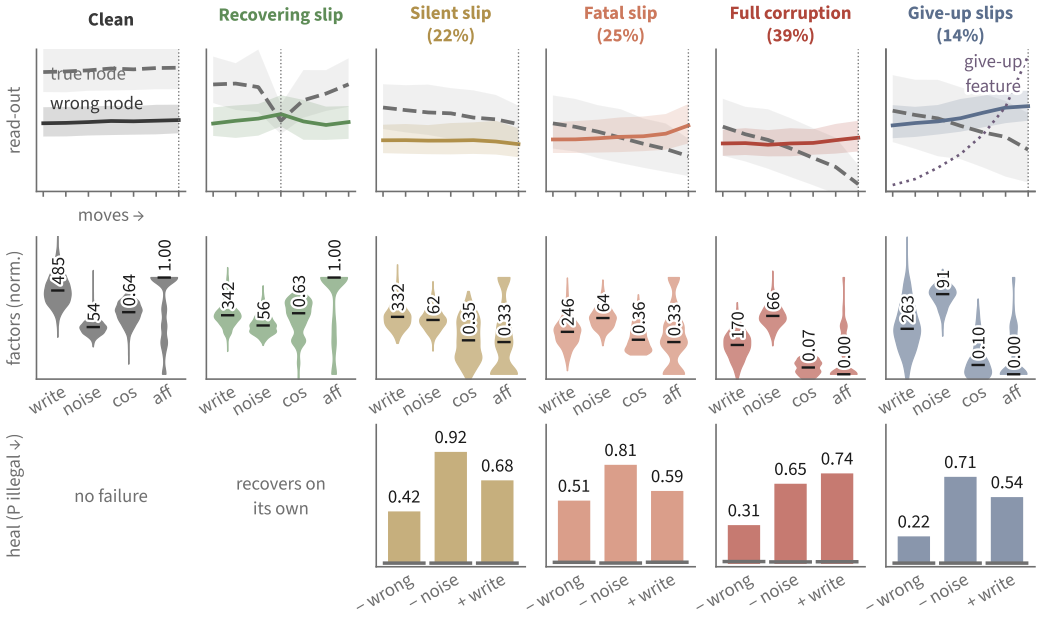


Fig. 33: **Compression rides exhibit the same failure modes: weak position writes and noise, with recovery under the same position edits.** Columns compare failure categories and reference states. Rows show position readouts around failure or recovery, scaled write/noise/cosine/affordance measurements, and the fraction of illegal probability removed. Gray marks show matched random edits.

illegal moves and less agreement between prefixes. This supports the interpretation that compression largely tracks the same localization failures seen in the stress and detour tests. For the illegal-move rate, we stop counting at the first illegal move, including that move. Continuation length includes all generated moves, even those after an illegal move. The upper block groups the 146 sampled-prefix trials by goal distance.

Continuous position reinforcement. For the reinforcement experiment, we add the true intersection's difference-in-means vector at layer 11 after each move token ($\alpha = 1$), using the large random-walk model. The layer is selected by the minimal teleportation test. We obtain the true position by following

Table 12: **Compression scores fall as the goal becomes more distant.** Upper block: 146 supplementary trials with sampled prefix lengths. Lower block: the same 60 pairs of 12-move routes in each band, changing only the goal (seed 0). Illegal /1k is the off-graph rate per thousand checked moves. Decode requires both prefixes to identify the shared intersection. Dashes indicate unrecorded continuation lengths.

		median length (moves)				
goal distance (moves)	pairs	route	continuation	illegal /1k	diff-means decode	score
sampled-prefix diagnostic run						
1–20	18	25.5	–	0.16	100%	0.722
21–40	45	29	–	0.89	100%	0.622
41–60	49	23	–	4.69	100%	0.184
61+	34	9.5	–	8.76	100%	0.059
controlled goal-swap experiment: fixed routes, varying destination						
1–4	60	12	17	0.06	100%	0.983
5–11	60	12	43	0.10	100%	1.000
12–20	60	12	62	0.11	98.3%	0.933
21–30	60	12	72	0.27	100%	0.817
31–45	60	12	83	1.18	100%	0.700
46–60	60	12	86	4.45	100%	0.400
61–99	60	12	85	9.18	100%	0.167

the actual moves and stop editing after an illegal transition. For compression, we apply the edit both when generating continuations and when scoring them under the other prefix. Detour and compression use the original benchmark implementations. We reuse the 20,000 baseline stress pairs and evaluate 1,000 detour cases and 1,936 scored compression pairs from 2,500 attempts (seed 0).

H Mechanistic indicators across models and through training

H.1 Across models

Feature and layer selection. We fit intersection centroids on each model's training rides with a budget of 100 states per intersection, evaluate on its held-out rides, and select layers by that model's sweeps. The decode indicator is the share of intersections read correctly in at least 90% of their held-out states. Teleportation selects the layer on 200 scenes, then reports the share of targets succeeding on more than half of all their eligible one-move scenes. The original benchmark code supplies compression and detour scores. Table 13 collects all comparison metrics, including street steering and position decoding after removing past position codes. For street steering, we inject the source intersection feature at the preceding position at every layer and sweep all read layers. Street steering selects the injection-strength grid and read layer with the best top-one score using fitted response slopes across strengths. At that choice, the reported top-five accuracy uses the endpoint increase.

Street-probe control. To check whether low street-probe accuracy reflects difficulty reading the current position, we also train probes to predict the current intersection. Both tests use one linear map per move and hold out a quarter of source intersections. Current-position accuracy is above 99% on the random-walk models, but only 50.3% and 33.2% on SP and NSP. For the latter models, even the current position is therefore harder to read at held-out intersections.

Feature-extraction and teleportation controls. On the same teleportation scenes, success rises from 3.8% to 26.8% on SP and from 6.3% to 31.5% on NSP when we apply the position edit. On the three random-walk models, it rises from below 0.1% to 82.2–87.4%. These rates average over scenes, rather than counting the intersections that pass the test as in Table 13. We also compare diff-means with probe directions and whitened means in a separate SP experiment. The probe gives better held-out decoding than diff-means (92.3% versus 53.9%), but lower one-move teleportation success at the selected steering layer (20.5% versus 35.5%; 2.5% without an edit). Whitened means give 69.6% decoding and 4.5% teleportation success. Better decoding alone therefore does not make a direction better for steering.

Table 13: **World-modeling measurements across models.** `SP`, `NSP` and `RW` denote shortest paths, noisy shortest paths and random walks; `NTP` and `NextLat` denote next-token and next-latent prediction. Street steering reports top-five next-intersection accuracy; tracking reports current-position accuracy before and after removing past position codes. Selected layers appear below each score.

	Map / localization							Navigation		Behavior		
	decode	causal	streets steer	streets probe	tracking	super-position	by afford.?	legal moves	goal compass	stress test	detour test	compr.
	% int $\geq .9$	telep $>50\%$	top5	held out	$-$past pos	angle / dims	own class	% int.	steer err	on graph	success	score
`SP·NTP`	12.7%	14.9%	8.2%	23.5%	.19→.09	35.9°	41.0%	18.1%	25.8°			
12L×768d×12h	L6	L10	L8	L7	L6	162d, L6	L6	L10	L12	71.7%	0.0%	.101
`NSP·NTP`	17.1%	16.9%	8.0%	12.2%	.21→.02	38.0°	41.1%	22.3%	23.2°			
48L×1600d×25h	L43	L46	L45	L36	L43	251d, L43	L43	L44	L47	74.7%	0.2%	.054
`RW·NTP`	99.6%	94.0%	93.5%	89.2%	1.00→.15	48.1°	91.4%	99.4%	18.1°			
48L×1600d×25h	L18	L11	L17	L15	L18	244d, L18	L18	L35	L15	91.5%	63.1%	.524
`RW·NTP`	99.7%	88.5%	39.7%	81.9%	1.00→.11	47.1°	88.2%	96.9%	17.3°			
48L×384d×8h	L40	L31	L21	L39	L40	210d, L40	L40	L44	L43	96.7%	77.5%	.523
`RW·NextLat`	99.9%	87.1%	17.5%	94.8%	1.00→.17	42.2°	84.7%	99.0%	17.6°			
48L×384d×8h	L36	L27	L21	L44	L36	183d, L36	L36	L44	L37	97.3%	78.5%	.556

Legal-move measurement and baseline. We normalize probabilities over the eight move tokens and classify moves above 3% as legal. A prediction is correct only if the complete legal-move set matches. Always guessing the most common set scores 9.6%. The single-model analysis uses a 1% threshold; the cross-model comparison and training sweep use 3%. The layer sweep excludes the last two layers of SP and the last four layers of the 48-layer models; ties use the earliest layer.

Position-tracking control. We compare removing past position information with removing a random subspace of the same dimension. Across the five models, random-subspace removal changes decoding accuracy by at most 0.4 percentage points, whereas removing past position information lowers it by 9.6–89.0 percentage points.

Training details. We use the authors' released checkpoints. Table 14 summarizes their training budgets. Vafa et al. (2024) train SP until overfitting and select the best validation checkpoint; for NSP and RW, they use the last validation checkpoint after five and one epochs, respectively. Teoh et al. (2026) train for six epochs because performance does not generally converge within one epoch. The smaller RW models' stronger performance may therefore partly reflect their longer training.

Table 14: **Reported training budgets of the compared models.** Corpus sizes and epochs follow Vafa et al. (2024) and Teoh et al. (2026). Total training tokens are corpus tokens multiplied by epochs, counting repeated passes. SP uses the best validation checkpoint, so its epoch count and total exposure are unspecified. Token counts are in billions. Global batch sizes count rides for Vafa models and packed 256-token sequences for small RW models.

Model	Batch size	Corpus tokens (B)	Epochs	Training tokens (B)
SP	48	~ 0.12	—	—
NSP	48	~ 1.68	5	~ 8.39
Large RW	48	~ 4.74	1	~ 4.74
Small RW·NTP	256	~ 4.74	6	~ 28.41
Small RW·NextLat	256	~ 4.74	6	~ 28.41

Slip and recovery measurement. For $1{,}500$ stress rides per random-walk model, we rank mean-centered unit intersection directions at each state, using each model's selected layer. A slip is an incorrect top-ranked node; recovery means a correct readout within five further moves. If the ride ends before recovery and before five further moves, an illegal ending counts as a failure to recover; other endings are excluded from the recovery rate. Table 15 reports both rates, using the same direction-based position readout as the failure analyses.

Table 15: **Smaller random-walk models slip less often under stress.**

model		slips % of steps	recovers within 5 steps
RW·NTP	48L×1600d	1.52%	67.8%
RW·NTP	48L×384d	0.56%	91.2%
RW·NextLat	48L×384d	0.24%	77.4%

H.2 THROUGH TRAINING

Training and measurement details. The model uses Teoh et al.'s (2026) 48-layer, 384-wide architecture with eight heads. We train on random walks with next-token prediction, context length 256, effective batch size 256, seed 1234, and Adam at learning rate 10^{-4} for 300,000 updates. Each indicator uses the fixed layer shown in Table 16, selected from the finished run. The table gives the raw values underlying Fig. 7, with controls, at twelve checkpoints spanning the run. The figure normalizes each indicator to its final value; for the compass, we normalize the reduction in angular error from the random baseline of 90°. We report the share of intersections decoded at both 90% and 50% accuracy. For street steering, we inject position features at every layer from 1 through 24 and rank next-intersection features at layer 24. We also measure teleportation without an edit, edit strength relative to the residual norm, current-intersection probing, and tracking after random-subspace removal. A different-move street-steering control gives lower top-five accuracy at every measured checkpoint.

The street probing test is a more reliable indicator than the street steering test. Between steps 30,000 and 300,000, street steering top-five accuracy falls from 59.3% to 32.2%, while street probing accuracy gradually rises from 72.4% to 82.6% (Table 16). The latter is more consistent with improving street encoding. NextLat also performs poorly on the steering test despite strong probing performance (Table 13), even though its training objective encourages learning transitions. These results lead us to prefer the street probing test as an indicator. We treat strong steering performance as sufficient, but not necessary, evidence of street encoding. The steering test uses averaged residual streams and supplies only one previous position, rather than the look-back window the model normally uses. We hypothesize that changes in how the model uses past positions during training can make this artificial setup less effective, even as street connections become more accurately linearly decodable.

A high compression score need not imply successful navigation. Even a model that navigates poorly can score highly on compression, because both prefixes can agree on incorrect continuations. At step 200, compression is 0.404, although only 1.8% of stress rides remain on the graph. A high compression score is therefore informative only when the model also performs reasonably well on the task. Fig. 7 shows compression from the fourth measured checkpoint (step 1,500); Table 16 retains all measurements.

Table 16: **World-modeling capacities develop at different stages of training.** Raw measurements at twelve checkpoints of the small RW·NTP model, using fixed layers. The upper block covers localization and street encoding; the lower block covers packing, navigation, and behavior. Legal-move prediction and the goal compass improve before reliable position decoding. For causal measurements, floor is teleportation success without an edit and dose is the edit-to-residual norm ratio. Tracking compares the full history with removal of past position information ($-$pos) or a random-subspace edit ($-$rand); street probes predict the next or current intersection. Packing reports the nearest-neighbor feature angle, the dimensions explaining 90% of variance (d_{90}), and the percentage of features closest to their own legal-move group's mean (affordance). Accuracy and success rates are percentages, except causal mean, tracking, and legal-set accuracy, which are fractions. Compression is also a fraction; angle and compass error are in degrees. Dashes mark unavailable measurements.

	Decode (L40)		Causal (L33)				Tracking (L40)			Streets: steering test (L24)		Streets: probe test (L39)	
step	$\geq .9$	$\geq .5$	$> .5$	mean	floor	dose	full	$-$pos	$-$rand	top1	top5	next	current
50	0.3	0.3	8.6	0.31	13.6	0.28	–	–	–	–	–	–	–
200	0.4	1.4	12.4	0.37	9.2	0.64	0.03	0.02	0.04	0.6	2.0	1.6	2.2
700	3.8	20.4	26.5	0.48	10.8	1.03	0.27	0.03	0.27	7.8	18.9	12.3	22.8
1500	8.8	43.5	32.6	0.51	10.0	1.07	0.51	0.02	0.47	10.4	25.3	20.5	43.6
3000	21.8	80.4	50.3	0.59	6.8	1.08	0.71	0.01	0.64	10.8	27.6	34.4	65.4
5000	40.4	96.6	57.9	0.64	3.2	1.11	0.83	0.02	0.76	13.3	32.8	46.2	78.9
10000	71.1	99.6	53.1	0.61	0.4	1.11	0.89	0.02	0.86	24.5	52.8	59.4	90.4
30000	95.1	100.0	66.7	0.68	0.0	1.12	0.98	0.06	0.94	28.3	59.3	72.4	97.5
45000	97.6	100.0	66.7	0.69	0.0	1.12	0.99	0.05	0.97	25.6	56.1	74.9	98.2
110000	99.1	100.0	77.9	0.75	0.0	1.09	1.00	0.07	0.98	19.9	46.5	79.3	98.9
200000	99.3	100.0	82.6	0.80	0.0	1.08	1.00	0.08	0.99	10.7	28.3	81.0	99.0
300000	99.6	100.0	88.1	0.84	0.0	1.07	1.00	0.11	0.99	12.8	32.2	82.6	99.3

	Packing (L40)			Navigation		Behavioral tests		
step	angle	d_{90}	affordance	legal set (L44)	compass (L43)	stress on-graph	detour arrival	compression
50	4.4	1	32.2	0.024	86.9	0.5	0.0	0.000
200	11.9	5	45.4	0.025	77.7	1.8	0.0	0.404
700	21.8	34	62.6	0.386	32.9	2.9	0.0	0.025
1500	25.1	76	87.1	0.868	24.6	3.3	4.0	0.000
3000	29.3	115	96.4	0.980	20.5	–	–	–
5000	33.5	142	97.7	0.993	20.3	–	–	–
10000	38.6	167	98.3	0.994	19.4	48.5	29.0	0.081
30000	44.3	189	96.0	0.993	18.6	–	–	–
45000	45.5	194	95.2	0.994	19.1	87.0	59.5	0.329
110000	46.4	199	92.2	0.992	18.3	–	–	–
200000	46.5	203	90.1	0.989	18.5	–	–	–
300000	46.4	204	89.0	0.988	18.3	95.0	77.0	0.484