KNOWING WHAT TO CHANGE IS NOT ENOUGH: STATE-TRANSITION LOCALITY IN FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Foundation models increasingly update persistent structured state from new evidence. A reliable transition must change the variables supported by the evidence and preserve the rest. Across 205,920 controlled outputs from 13 open foundation models, these capabilities diverge: models with near-perfect target correctness can still have poor complement retention. Qwen3.8-27B, for example, reaches 99.5% joint target correctness but only 29.7% conditional retention. We formalize this failure as *update spillover* and separate support recovery, value prediction, and complement preservation. We then run a 2×2 controlled interface intervention across ten models, varying whether transition support is predicted or provided and whether the model emits a full vector or a local update. Predicted sparse deltas underperform free full-vector decoding for nine of ten models. The oracle-local interface preserves the complement by construction and yields 86.5–100% exact transition accuracy. Explicit deterministic slot–value evidence nearly closes the gap. The hard regime is semantic support inference followed by scoped update execution, making transition locality a distinct reliability criterion for models that maintain persistent state.

1 INTRODUCTION

Agent systems increasingly couple foundation models to external actions, memories, and persistent records (Yao et al., 2023; Park et al., 2023; Packer et al., 2023). Updating such state is a basic operation: new evidence may change one part of a task, environment, or record while the rest should remain fixed. Reliable updating is therefore local—change what the evidence supports and preserve everything else.

A full-state interface combines three decisions in one generation: which variables are supported to change, what their new values should be, and whether unsupported variables remain unchanged. Correct target prediction does not guarantee that the surrounding state survives regeneration. Standard next-state accuracy merges these decisions and can hide collateral changes.

We observe this separation at scale. In controlled eleven-dimensional updates, several strong models identify both the target variable and update direction in more than 95% of cases, yet preserve every unsupported field in fewer than one third of those correct cases. We call unsupported changes *update spillover* and formalize the gap between target correctness and complement preservation as *state-transition locality*.

Our evaluation has two layers. First, we reanalyze 205,920 existing outputs from 13 open foundation models to measure the locality gap, its field structure, and its uncertainty. Second, we intervene on the update interface while holding the model, evidence, examples, and decoding settings fixed. A 2×2 design crosses predicted versus oracle support with full-vector versus local output, separating support routing from update execution.

The intervention yields a consistent pattern. Oracle support improves full-vector updates for every model, while complement errors remain for several models. Predicted sparse deltas are worse than free full-vector decoding for nine of ten models. When oracle support is paired with a local value interface, all ten models operate through an interface that structurally preserves the complement and

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.

reach 86.5–100% exact transition accuracy. A deterministic slot–value control nearly closes the gap, placing the difficult regime in semantic support inference followed by scoped execution.

Our contributions are:

- We formulate state-transition locality for persistent structured-state updating and introduce conditional metrics that separate target correctness, support recovery, and complement preservation.
- We establish the locality gap across 13 open foundation models and show recurring field-specific spillover structure.
- We identify support routing and update execution as distinct bottlenecks through a four-interface intervention, with deterministic-evidence, instruction-strength, serialization, and paraphrase controls defining the regime in which the gap persists.

2 Related Work

Persistent agent state and memory. LLM agents increasingly act through external environments and persistent memory. ReAct interleaves reasoning with external actions (Yao et al., 2023), while Generative Agents and MemGPT maintain and retrieve experience across long interactions (Park et al., 2023; Packer et al., 2023). We study the state transition itself: given a previous structured state and new evidence, which variables should change and which should remain fixed.

Locality in model editing. Model editing evaluates whether a targeted change generalizes to its intended scope while preserving unrelated behavior. MEND, ROME, and MEMIT edit model parameters (Mitchell et al., 2022a; Meng et al., 2022; 2023); SERAC instead routes edits through an explicit memory and learns their scope (Mitchell et al., 2022b). Our setting keeps the foundation model frozen and measures locality in inference-time updates to an external state.

Dialogue state tracking and selective updating. Dialogue state tracking provides the closest operational precedent for structured incremental updates. TRADE and TripPy generate or copy slot values (Wu et al., 2019; Heck et al., 2020); SOM-DST explicitly predicts slot operations before selectively overwriting memory (Kim et al., 2020); Correctable-DST detects slots that require correction before generation (Xie et al., 2022). Schema-guided DST makes the slot schema itself an input (Rastogi et al., 2020), and recent work applies general-purpose LLMs through instruction tuning or function calling (Feng et al., 2023; Li et al., 2024). These systems establish selective slot updates as a useful design pattern. We evaluate whether frozen foundation models recover and obey update scope under a controlled, domain-independent transition interface.

Constrained structured generation. Grammar- and schema-constrained decoding can enforce syntactic structure (Geng et al., 2023). Recent benchmarks show that constraint compliance, generation quality, and decoding efficiency remain distinct axes (Geng et al., 2025), while schema-key wording can itself alter model behavior (Le, 2026). Our protocols enforce protocol-specific JSON schemas and score semantic transition scope after parsing: which fields changed, whether the target changed correctly, and whether unsupported fields were preserved.

3 State-Transition Locality

3.1 Persistent State Updating

Let $\mathbf{s}_t = (s_t^1, \ldots, s_t^K)$ be a persistent structured state and e_t new evidence. A transition produces $\mathbf{s}_{t+1}$. The true transition support is

$$S^* = \{k : s_{t+1}^k \neq s_t^k\}, \tag{1}$$

and the support induced by a model output is $\hat{S}$. In our controlled single-target edits, $|S^*| = 1$.

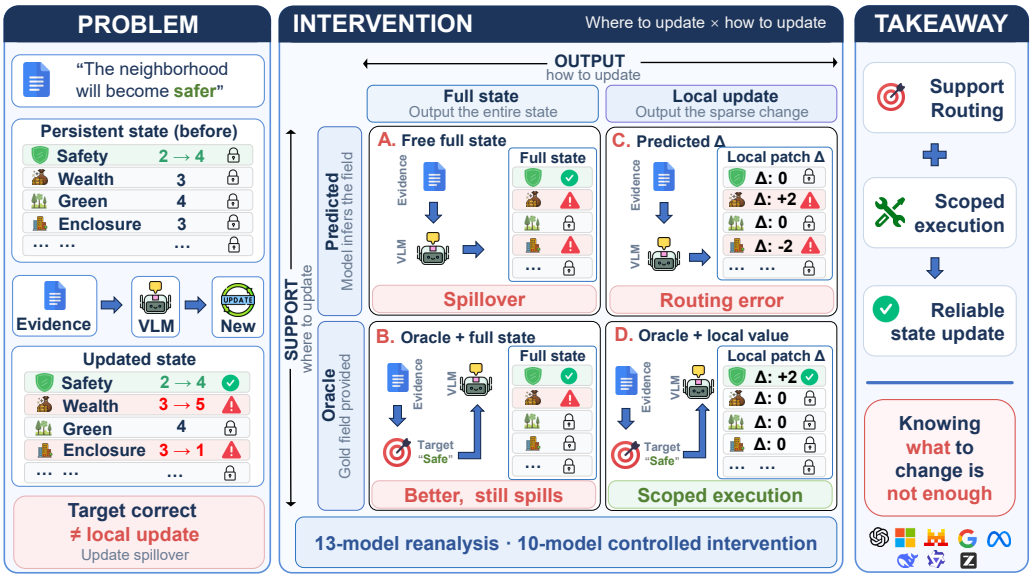


Figure 1: Two-factor intervention on transition support and output scope. Rows set the support source; columns set the output scope. All other inputs and decoding settings are fixed.

3.2 Support and Spillover

A local transition changes only variables in the evidence-supported transition. Unsupported predicted changes form the spillover set

$$U_{\text{spill}} = \hat{S} \setminus S^*, \qquad C_{\text{spill}} = |U_{\text{spill}}|. \tag{2}$$

3.3 Transition Metrics

Support precision, recall, and F1 compare $\hat{S}$ with S^*. Our central conditional retention rate is

$$\text{CRR} = P(C_{\text{spill}} = 0 \mid \text{joint target correct}), \tag{3}$$

while conditional spillover count is

$$\text{CSC} = E[C_{\text{spill}} \mid \text{joint target correct}]. \tag{4}$$

We also report the unconditional any-spill rate

$$\text{ASR} = P(C_{\text{spill}} > 0), \tag{5}$$

which measures how often an update changes at least one unsupported field. Exact transition accuracy (ETA) requires the correct target action and no unsupported change. Transition edit distance (TED) counts target-action error plus unsupported changed fields.

4 A 2 × 2 Interface Intervention

We intervene on two decisions in structured state updating: where to update and how much state to regenerate. Figure 1 crosses predicted versus oracle support with full-vector versus local output. Examples, checkpoint, evidence, decoding settings, and parser are fixed across protocols.

A: Free full vector. The model infers support and emits actions for all K fields.

B: Oracle support + full vector. The true support is supplied and the model emits all K actions.

C: Predicted sparse delta. The model predicts changed fields and their actions; the delta is deterministically applied to the previous state.

D: Oracle support + local value. The true support is supplied and the model predicts only the target action.

A→B measures the effect of support specification under full-vector output. A→C measures the effect of sparse output when support is still predicted. B→D measures local execution once support is known. C→D measures support specification under a local interface. We score target-action accuracy, support F1, ETA, TED, output burden, and parse failures.

5 Experimental Setup

5.1 FoundationProbe and Controlled Updates

FoundationProbe v3 provides controlled transitions with exactly known targets. It contains 15,840 outputs per model over eleven interpretable state fields, three evidence formats, bidirectional extreme edits, paraphrases, eight cities, and repeated base states. Across 13 models this yields 205,920 outputs. A deterministic monotone edit operator defines the target transition, making support and spillover directly measurable.

5.2 Models and Statistical Units

We evaluate thirteen open models spanning Qwen, Gemma, DeepSeek, GLM, Phi, Llama, Mistral, and GPT-OSS families. The four-interface experiment uses ten locally frozen GGUF checkpoints: Qwen3.8-27B, Qwen3.5-35B-A3B, DeepSeek-R1-Distill-Qwen-32B, DeepSeek-R1-0528-Qwen3-8B, Gemma-3-27B, Gemma-4-12B, GLM-4-9B-0414, GPT-OSS-20B, Phi-4-Multimodal, and Phi-4-Reasoning-Vision-15B. Checkpoints use their released Q4 variants (Q4_K_M, Q4_K_XL, or MXFP4) and are served with the CUDA build of `llama.cpp`. We decode at temperature zero and top-$p = 1$ under a strict protocol-specific JSON schema. Context windows, reasoning modes, token budgets, repository identifiers, and exact artifact filenames are frozen in the model manifest. Confidence intervals resample `base_id` clusters with 2,000 bootstrap replicates and seed 2027.

5.3 Four-Protocol Sample

The controlled interface experiment uses 550 unique base states: 50 per field, with 275 increasing and 275 decreasing transitions. The same model checkpoint, evidence, decoding parameters, parser, examples, and item IDs are used in all four protocols; only support information and output representation change.

6 The Locality Gap Across 13 Models

6.1 Target Accuracy and Retention Diverge

Figure 2 shows a two-axis capability profile under the single-sentence-in-context protocol. Model-level joint target correctness and CRR have Spearman correlation -0.17 with 95% cluster-bootstrap CI $[-0.26, -0.10]$. Qwen3.8-27B reaches 99.5% joint correctness but only 29.7% CRR; Gemma-3-27B reaches 95.9% and 31.9%; DeepSeek-R1-Qwen-32B reaches 91.3% and 80.2%. Target accuracy and transition locality are therefore distinct model properties.

6.2 Correct Targets Still Spill

Conditioning on a correct target isolates complement preservation from target recognition. Figure 3 shows that several high-accuracy models still change multiple unsupported fields. Complement corruption remains a major transition error after the intended target is correct.

6.3 Spillover Has Structure

The target-to-spill matrix in Fig. 4 is strongly non-uniform. Spillover concentrates along recurring field relationships, including safety–pedestrian infrastructure and liveliness–boredom. Unsupported changes follow reproducible field-specific patterns.

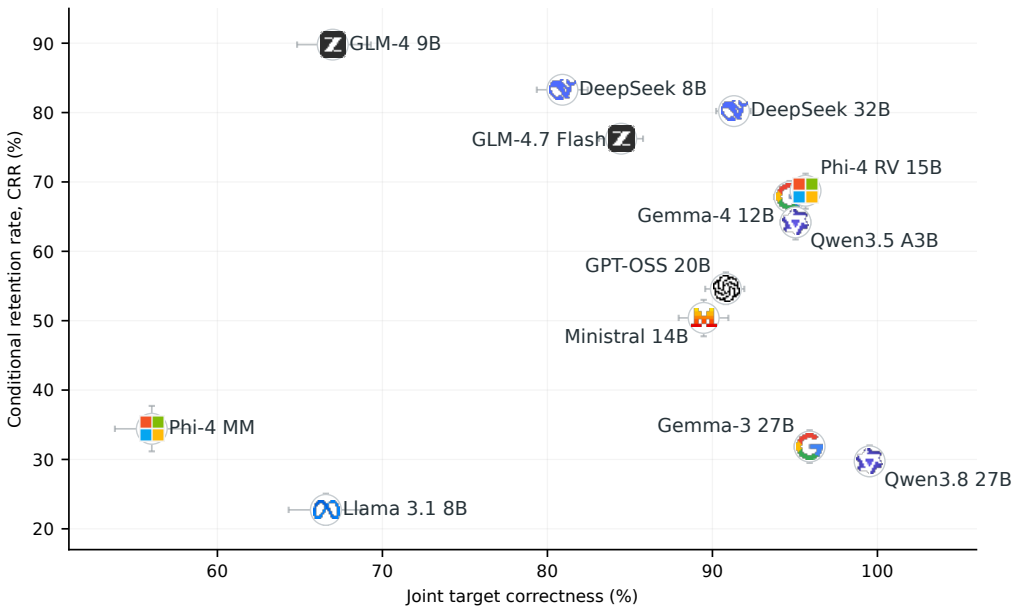


Figure 2: Target correctness and complement retention diverge across 13 foundation models. Error bars are 95% cluster-bootstrap intervals over base states.

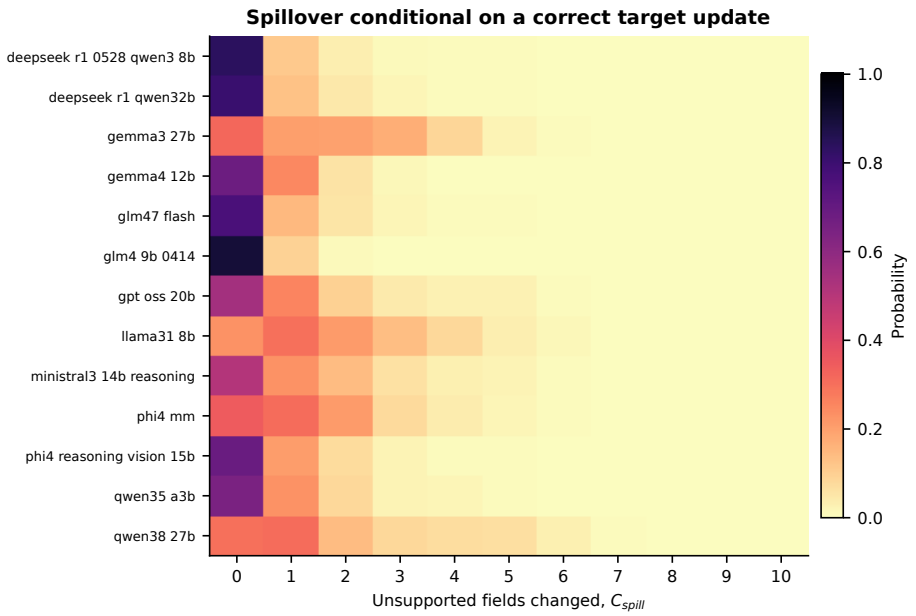


Figure 3: Unsupported fields still change after conditioning on a correct target update.

Field-wise results show the same separation. Averaging model-level metrics within the `single_sentence_in_context` protocol, `wealthy` reaches 90.1% joint target correctness but 38.7% CRR, whereas `road_surface` reaches 91.5% and 87.0%. Across the eleven fields, the descriptive Spearman association between mean joint correctness and CRR is $\rho = 0.45$ ($p = 0.17$). Recognition difficulty and retention difficulty do not coincide across fields.

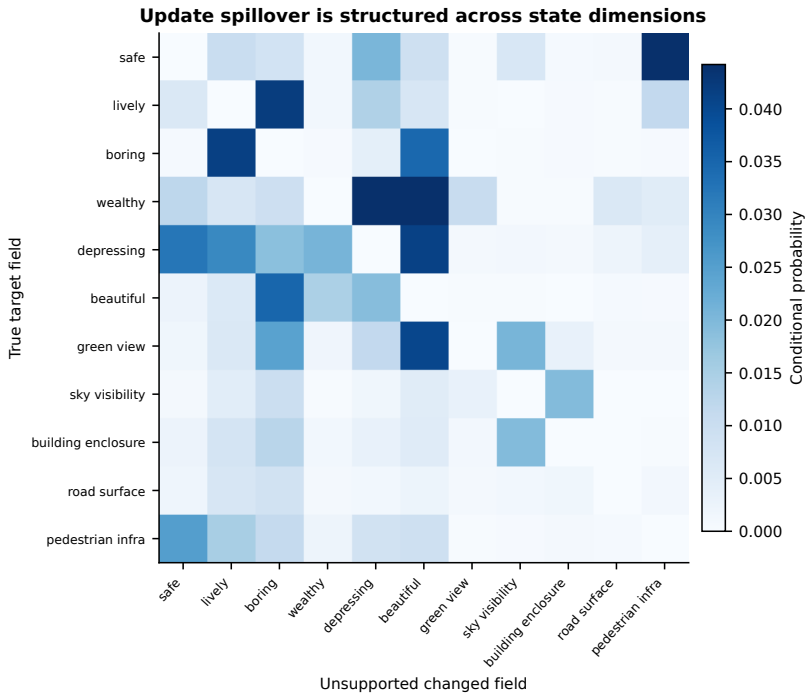


Figure 4: Update spillover follows a structured target-to-field topology. The true-support diagonal is excluded.

Table 1: Exact transition accuracy (%) for the four controlled interfaces. Each cell contains 550 unique base states.

Model	A	B	C	D
Qwen3.8-27B	75.6	99.8	54.5	**100.0**
Qwen3.5-35B-A3B	66.7	93.3	43.5	**97.8**
DeepSeek-R1-Qwen-32B	68.2	92.9	46.9	**93.6**
DeepSeek-R1-Qwen3-8B	44.7	65.5	25.8	**91.1**
Gemma-3-27B	52.4	79.6	42.4	**99.5**
Gemma-4-12B	65.1	88.5	45.8	**100.0**
GLM-4-9B	62.7	63.3	34.4	**87.6**
GPT-OSS-20B	70.2	84.0	44.9	**98.5**
Phi-4-MM	10.4	37.5	11.1	**86.5**
Phi-4-Reasoning-15B	58.7	60.2	40.2	**94.0**

7 What Causes the Gap? Controlled Interface Interventions

7.1 Support Routing and Update Execution Both Matter

Oracle support raises full-vector ETA for all ten models. The gains are large for Qwen3.8, DeepSeek-32B, Gemma-3, and Phi-4-MM, while several models still make complement errors under protocol B. Support specification therefore removes one source of error without eliminating full-vector spillover.

Output scope supplies the second factor. Protocol C underperforms A for nine of ten models, with Phi-4-MM changing only from 10.4% to 11.1%; sparse output does not repair support routing. Protocol D supplies the support and predicts only the target value, preserving the complement by construction. ETA reaches 86.5–100% across all ten models. Reliable local updating requires correct support routing and scoped execution.

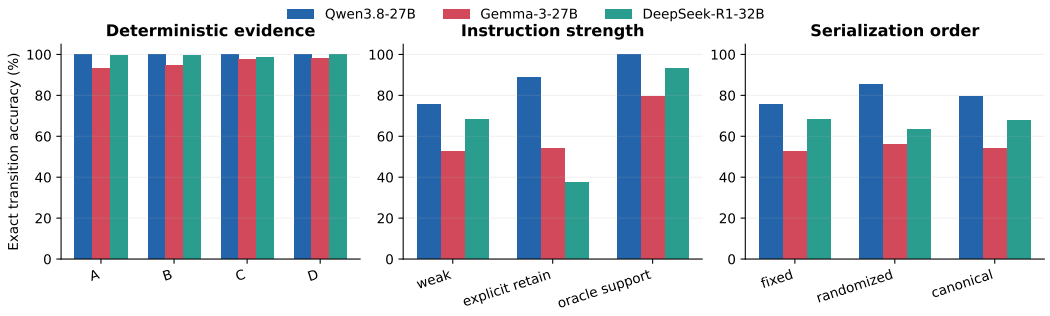


Figure 5: Boundary and robustness controls across three model families. Explicit deterministic evidence nearly closes the locality gap; retention wording and serialization have model-dependent effects.

8 When Does the Gap Persist?

Generator-independent evidence. Explicit deterministic slot–value evidence nearly closes the gap. Qwen3.8 is perfect under A–D; free-full-vector ETA reaches 99.6% for DeepSeek-32B and 93.3% for Gemma-3, while oracle-local D reaches 100.0% and 98.0%. Semantic support inference is therefore the difficult regime captured by the main experiment.

Instruction strength. Retention wording has model-dependent effects. Qwen3.8 rises from 75.6% to 88.7% (+13.1 pp, 95% CI [+9.1, +16.7]); Gemma-3 changes from 52.4% to 53.8% (+1.5 pp, 95% CI [-2.4, +5.5]); DeepSeek-32B falls from 68.2% to 37.3% (-30.9 pp, 95% CI [-36.0, -25.8]). Oracle support reaches 99.8%, 79.6%, and 92.9% on the same models, giving a consistent improvement over the weak prompt.

Serialization. Field order also has model-dependent effects. Randomizing field order raises Qwen3.8 from 75.6% to 85.3% (+9.6 pp, 95% CI [+5.8, +13.6]), changes Gemma-3 from 52.4% to 56.0% (+3.6 pp, 95% CI [-0.2, +7.6]), and lowers DeepSeek-32B from 68.2% to 63.3% (-4.9 pp, 95% CI [-9.5, -0.4]). Canonical-order changes are small for all three. No serialization order improves all three models.

Paraphrase stability and edit magnitude. Support routing varies across surface forms. Across three paraphrases of the same transition, mean pairwise Jaccard similarity ranges from 70.9% to 90.9% across the 13 models. FoundationProbe uses only extreme ordinal transitions ($0 \leftrightarrow 4$), so this experiment does not identify edit-magnitude effects.

9 Discussion

Transition locality is a separate capability from target prediction. The 13-model reanalysis shows that high target accuracy can coexist with low complement retention, and the spillover topology shows recurring field-specific error structure. A single next-state score would merge these behaviors.

Support routing and output scope interact. Oracle support improves full-vector updates, but full-state regeneration still produces complement errors for several models. Sparse output becomes reliable only when the correct support is available. Exposing transition support as an explicit interface object therefore improves the controllability of state updates.

The deterministic control localizes the main difficulty to semantic support inference. When the changed slot and value are stated almost literally, all four interfaces are highly reliable in the three tested models. Richer evidence requires the model to infer which state variable the evidence licenses before executing the change, and this regime produces the largest locality differences.

Our evaluation uses eleven urban dimensions and single-target edits, which make support exactly scoreable. Broader schemas, multi-target transitions, ambiguous evidence, and multimodal observations are the next settings for testing the same decomposition.

10 Conclusion

Reliable state updating requires both the correct change and the correct scope. Across 13 open foundation models, target correctness and complement preservation diverge, and unsupported changes follow structured field relationships. Across ten models, the controlled interface intervention separates support routing from update execution: predicted sparse output remains unreliable, while oracle support paired with a local value interface yields 86.5–100% ETA and preserves the complement by construction. Explicit slot–value evidence nearly closes the gap. State-transition locality measures whether a model can turn semantically expressed evidence into a scoped update while keeping persistent state intact.

AI use statement

In this work, a Qwen-family model generated the original textual evidence used in FoundationProbe. Generative AI tools also assisted with experimental-design feedback, visualization code, literature discovery and language editing. The authors ran the experiments, verified reported statistics against machine-readable artifacts, checked cited sources, reviewed AI-assisted code and text, and made the final scientific and editorial decisions. We take responsibility for all text, claims, code, figures, and other artifacts in this submission.

Ethics statement

The experiments study model behavior on controlled structured-state edits. Urban attributes serve as interpretable test dimensions; the reported outputs are not used to make decisions about individuals or communities, and the paper makes no causal or normative claims about real locations.

Reproducibility statement

Metric definitions, statistical units, model and decoding configurations, and the controlled intervention are described in the main paper. The appendix provides exact checkpoints, prompts, schemas, parsing rules, data construction, and bootstrap settings.

References

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. Towards llm-driven dialogue state tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 739–755, 2023. doi: 10.18653/v1/2023.emnlp-main.48.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured nlp tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10932–10952, 2023. doi: 10.18653/v1/2023.emnlp-main.674.

Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. Generating structured outputs from language models: Benchmark and studies. *arXiv preprint arXiv:2501.10868*, 2025.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gašić. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 35–44, 2020. doi: 10.18653/v1/2020.sigdial-1.4.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 567–582, 2020. doi: 10.18653/v1/2020.acl-main.53.

Yifan Le. Schema key wording as an instruction channel in structured generation under constrained decoding. *arXiv preprint arXiv:2604.14862*, 2026.

Zekun Li, Zhiyu Zoey Chen, Mike Ross, Patrick Huber, Seungwhan Moon, Zhaojiang Lin, Luna Dong, Adithya Sagar, Xifeng Yan, and Paul A. Crook. Large language models as zero-shot dialogue state tracker through function calling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8688–8704, 2024. doi: 10.18653/v1/2024.acl-long.471.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831, 2022b.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023. doi: 10.1145/3586183.3606763.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8689–8696, 2020. doi: 10.1609/aaai.v34i05.6394.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 808–819, 2019. doi: 10.18653/v1/P19-1078.

Hongyan Xie, Haoxiang Su, Shuangyong Song, Hao Huang, Bo Zou, Kun Deng, Jianghua Lin, Zhihui Zhang, and Xiaodong He. Correctable-dst: Mitigating historical context mismatch between training and inference for improved dialogue state tracking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 876–889, 2022. doi: 10.18653/v1/2022.emnlp-main.56.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

A Reproducibility Details

A.1 Model Card

Table 2 records the exact GGUF artifacts used in the 13-model reanalysis. The ten models evaluated in the controlled interface study are all rows except GLM-4.7-Flash, Llama-3.1-8B, and Ministral-3-14B.

Table 2: Frozen checkpoints used in the experiments. All models are served locally with the CUDA build of `llama.cpp`; no parameter update is performed. "Text" means that only the language interface of a multimodal checkpoint is used.

Model ID	Source repository	Exact model file	Interface
deepseek_r1_0528_qwen3_8b	unsloth/DeepSeek-R1-0528-Qwen3-8B-GGUF	DeepSeek-R1-0528-Qwen3-8B-Q4_K_M.gguf	Text
deepseek_r1_qwen32b	bartowski/DeepSeek-R1-Distill-Qwen-32B-GGUF	DeepSeek-R1-Distill-Qwen-32B-Q4_K_M.gguf	Text
gemma3_27b	bartowski/google_gemma-3-27b-it-GGUF	google_gemma-3-27b-it-Q4_K_M.gguf	Text
gemma4_12b	google/gemma-4-12B-it-qat-q4_0-gguf	gemma-4-12B-it-qat-UD-Q4_K_XL.gguf	Text
glm47_flash	ggml-org/GLM-4.7-Flash-GGUF	GLM-4.7-Flash-Q4_K_M.gguf	Text
glm4_9b_0414	unsloth/GLM-4-9B-0414-GGUF	GLM-4-9B-0414-Q4_K_M.gguf	Text
gpt_oss_20b	ggml-org/gpt-oss-20b-GGUF	gpt-oss-20b-MXFP4.gguf	Text
llama31_8b	bartowski/Meta-Llama-3.1-8B-Instruct-GGUF	Meta-Llama-3.1-8B-Instruct-Q4_K_M.gguf	Text
ministral3_14b_reasoning	mistralai/Ministral-3-14B-Reasoning-2512-GGUF	Ministral-3-14B-Reasoning-2512-Q4_K_M.gguf	Text
phi4_mm	microsoft/Phi-4-multimodal-instruct	phi4-mm-Q4_K_M.gguf	Text
phi4_reasoning_vision_15b	DevQuasar/microsoft.Phi-4-reasoning-vision-15B-GGUF	microsoft.Phi-4-reasoning-vision-15B.f16.gguf.Q4_K_M.gguf	Text
qwen35_a3b	Qwen/Qwen3.5-35B-A3B	Qwen3.5-35B-A3B-UD-Q4_K_XL.gguf	Text
qwen38_27b	ggml-org/Qwen3.8-27B-GGUF	Qwen3.8-27B-UD-Q4_K_M.gguf	Text

A.2 Generator-Family Split

The original evidence was authored with a Qwen model, so we report a family split alongside the generator-independent experiment. For the single-sentence-in-context probe, the eleven non-Qwen models average 83.0% joint target correctness, 75.4% Support F1, 60.0% CRR, 0.77 CSC, and 50.6% ETA. The two Qwen-family models average 97.3%, 74.3%, 46.9%, 1.14, and 45.3%, respectively. The deterministic-template experiment separately tests generator dependence.

A.3 Field-Wise Diagnosis

Table 3 averages model-level metrics over all thirteen models in the single-sentence-in-context setting. Target accuracy and complement retention vary differently across fields: `wealthy` reaches high joint target correctness but low CRR, whereas `road_surface` combines high target accuracy with high retention. Across the eleven fields, the Spearman correlation between target correctness and CRR is 0.45 ($p = 0.17$).

Table 3: Field-wise transition diagnosis averaged across 13 models. Accuracy, Support F1, CRR, and ETA are percentages; lower CSC and TED are better.

Field	Joint	F1	CRR	CSC	ETA	TED
beautiful	87.4	77.0	60.1	0.69	53.5	0.85
boring	65.9	70.5	55.6	0.70	35.5	0.98
building enclosure	83.4	80.4	71.5	0.53	59.3	0.69
depressing	72.5	63.3	49.3	1.52	34.9	1.67
green view	92.7	74.6	51.6	1.04	47.3	1.14
lively	91.4	76.8	51.4	0.85	48.4	0.87
pedestrian infra.	95.2	81.7	64.6	0.70	61.6	0.76
road surface	91.5	90.5	87.0	0.25	81.7	0.34
safe	71.9	60.1	39.0	1.08	23.8	1.29
sky visibility	95.0	86.8	71.8	0.43	68.7	0.48
wealthy	90.1	66.2	38.7	1.37	33.3	1.51

A.4 Inference and Parsing

All controlled protocols use temperature 0, top-$p = 1$, one completion per input, and a strict protocol-specific JSON schema. Context length and output budgets follow the frozen model registry; the largest reasoning budget is 1,024 tokens for DeepSeek-R1-Distill-Qwen-32B. The common system instruction requests the specified state-transition interface and JSON only. A parser searches the message content and, for reasoning checkpoints, the reasoning fields for the first complete JSON object. A malformed or exhausted response is retried once with a direct-answer system instruction and a 3,000-token minimum output budget. Attempt logs are append-only; metrics use unique successful item IDs, while schema failures are reported separately.

A.5 Data Card

FoundationProbe v3 contains 8,960 training states and 640 held-out base states from eight cities. Each state has eleven fields, five ordinal intensity levels, and three paraphrases. The reanalysis contains 15,840 outputs per model and 205,920 outputs in total, with repeated observations of shared base states. Statistical inference therefore resamples at the base-state level. The controlled interface study freezes 550 unique base states, balanced across the eleven target fields, and uses the same item IDs, checkpoints, decoding settings, and expected transition in every interface. The deterministic control replaces generator-authored evidence with five fixed templates per field while retaining the same scoring semantics.

Deterministic evidence templates. For each target field, old level, and new level, the five exact templates are: "New evidence sets {field} to {new}."; "The observed value of {field} is now {new}."; "For the updated state, record {field} as {new}."; "Only {field} has new evidence: its value is {new}."; and "The latest observation changes {field} from {old} to {new}." The five increasing level pairs are $(0,1),(0,2),(0,4),(1,3),(2,4)$; decreasing pairs reverse these endpoints. Crossing eleven fields, two directions, five level pairs, and five templates yields 550 unique controls.

Table 4: Exact transition accuracy (%) on generator-independent deterministic evidence. Each cell has 550 unique controls.

Model	A	B	C	D
Qwen3.8-27B	100.0	100.0	100.0	100.0
Gemma-3-27B	93.3	94.5	97.5	98.0
DeepSeek-R1-Qwen-32B	99.6	99.6	98.4	100.0

A.6 Statistical Protocol

Confidence intervals use 2,000 bootstrap replicates with seed 2027 and resample at the base-state level so paraphrases and protocol views of one base state are not treated as independent. Correlations are recomputed within each bootstrap sample. Field-wise and topology analyses condition spillover on a correct target update and exclude the true target from the spill set.

A.7 Complete 13-Model Diagnosis

Table 5 reports the complete single-sentence-in-context diagnosis underlying the main finding figure. Joint is correct target field and direction; CRR is complement retention conditioned on a correct target; CSC is the conditional spill count; ETA is exact transition accuracy; and TED is transition edit distance. High joint accuracy does not rank models by locality: Qwen3.8 has the highest joint score but one of the lowest CRRs.

Table 5: Full FoundationProbe transition-locality diagnosis. Accuracy, Support F1, CRR, and ETA are percentages; lower CSC and TED are better.

Model	Joint ↑	Support F1 ↑	CRR ↑	CSC ↓	ETA ↑	TED ↓
DeepSeek-R1-Qwen3-8B	80.9	81.7	83.3	0.26	67.7	0.47
DeepSeek-R1-Qwen-32B	91.3	86.3	80.2	0.32	73.4	0.40
Gemma-3-27B	95.9	64.8	31.9	1.60	30.5	1.62
Gemma-4-12B	94.7	83.7	67.8	0.42	64.2	0.45
GLM-4.7-Flash	84.5	82.5	76.2	0.42	65.5	0.53
GLM-4-9B	67.0	77.3	89.8	0.12	60.2	0.42
GPT-OSS-20B	90.8	77.3	54.6	0.87	49.6	0.99
Llama-3.1-8B	66.6	57.5	22.7	1.71	15.6	1.93
Ministral-3-14B-Reasoning	89.5	74.0	50.4	1.02	45.1	1.08
Phi-4-Multimodal	56.0	59.5	34.4	1.25	19.5	1.79
Phi-4-Reasoning-Vision-15B	95.6	84.9	68.7	0.47	65.8	0.51
Qwen3.5-35B-A3B	95.0	83.2	64.2	0.57	61.0	0.60
Qwen3.8-27B	**99.5**	65.5	29.7	1.70	29.6	1.70

A.8 Protocol Prompts

The prompt set comprises: (A) predicted support with a full transition vector; (B) oracle support with a full vector; (C) a predicted sparse delta; (D) oracle support with a local value update; (E) explicit retention; (F) deterministically randomized field order; and (G) canonical schema order. The experiment artifact records the prompt text, schemas, deterministic templates, parser, retry policy, frozen manifest, and scoring code.

A.9 Exact Prompt Templates

The following are the A–G user-prompt templates used by the runners. Braced names are replaced with frozen manifest values. The common system instruction is: "Follow the requested state-transition interface exactly. Return JSON only."

```
You maintain an eleven-field persistent state. Compare the previous observations with the new
    evidence.
Return one action for every field: "increase", "decrease", or "unchanged".
Also return "primary_dimension".
Use exactly the provided JSON schema and no extra text.

Previous observations:
{before_observations}

New evidence:
{after_observations}

JSON fields:
safe, lively, boring, wealthy, depressing, beautiful, green_view,
sky_visibility, building_enclosure, road_surface, pedestrian_infra,
primary_dimension
```

```
You maintain an eleven-field persistent state. The only state dimension directly supported to
    change is:
{target_field}

All other dimensions are unsupported by the new evidence and should be retained.
Return one action for every field: "increase", "decrease", or "unchanged".
Also return "primary_dimension". Use exactly the provided JSON schema and no extra text.

Previous observations:
{before_observations}

New evidence:
{after_observations}
```

```
Compare the previous observations with the new evidence. Predict the sparse transition support
     and values.
Return JSON with a single key "changes". Its value is a list of zero or more objects with keys
"field" and "action", where action is "increase" or "decrease". Do not emit unchanged fields.
Use no extra text.

Previous observations:
{before_observations}

New evidence:
{after_observations}
```

```
The only state dimension directly supported to change is {target_field}.
Predict its transition action from the previous observations and new evidence.
Return JSON with exactly two keys: "field" and "action". The field must be {target_field};
the action must be "increase" or "decrease". Use no extra text.

Previous observations:
{before_observations}

New evidence:
{after_observations}
```

```
You maintain an eleven-field persistent state. Update only state variables directly supported
    by the new evidence. Preserve every unsupported variable exactly as unchanged.
Return one action for every field: "increase", "decrease", or "unchanged".
Also return "primary_dimension". Use exactly the provided JSON schema and no extra text.

Previous observations:
{before_observations}

New evidence:
{after_observations}
```

```
You maintain an eleven-field persistent state. Compare the previous observations with the new
    evidence.
Return one action for every field: "increase", "decrease", or "unchanged".
Also return "primary_dimension". Emit the state fields in exactly this requested order:
{field_order}
Use exactly the provided JSON schema and no extra text.

Previous observations:
{before_observations}

New evidence:
{after_observations}
```

```
You maintain an eleven-field persistent state. Compare the previous observations with the new
    evidence.
Return one action for every field: "increase", "decrease", or "unchanged".
Also return "primary_dimension". Emit the state fields in exactly this canonical order:
{field_order}
Use exactly the provided JSON schema and no extra text.

Previous observations:
{before_observations}

New evidence:
{after_observations}
```