LEARNING A FACT IS NOT LEARNING HOW TO RETRIEVE IT

Anonymous authors
Paper under double-blind review

ABSTRACT

A model trained on `The capital of X is Y` may produce `Y` after `The capital of X is` but fail after `The capital of X:`. We call these different ways of eliciting the same fact *request forms*. To separate learning a fact from retrieving it, we train two models in two stages. In the first stage (request-form training), one model sees each fact in five forms and the other sees the same facts only as statements. In the second stage (target-fact training), both receive identical training on new facts, all as statements. Both then retrieve the new facts almost equally well from statements, but differ sharply on other request forms. Thus, a model can learn how to retrieve through a request form before it learns the facts. To understand this difference, we examine the hidden state immediately before the answer, which we call the *context state*. When given two different request forms for the same fact, the model trained on five forms in stage one produces more similar context states than the model trained on statements alone in that stage. Changing this state at retrieval time can enable or prevent retrieval of an already learned fact, and the same effect transfers across facts and factual relations, such as capitals and currencies. To test its role during learning, we change the context state only during target-fact training. This intervention changes later retrieval without intervention at test time. Together, these results show that later retrieval depends on earlier request-form experience and the context state during fact learning.

1 INTRODUCTION

Suppose we fine-tune a language model on `The capital of X is Y.` Later, the model produces `Y` after `The capital of X is` but fails after `- Capital of X:`. The fact can be retrieved through one form but not the other. We call these different ways of eliciting the same fact *request forms*. Prior work has observed related gaps between learning facts from text and retrieving them in a new form (Allen-Zhu & Li, 2024; Jiang et al., 2024). But such a failure leaves a basic ambiguity: did the model learn the fact only weakly, or did it fail to learn how to retrieve it through the second request form? In ordinary fine-tuning, the fact and the form in which it is presented are learned together, making these explanations difficult to distinguish.

We separate these possibilities with a two-stage experiment. In the first stage (request-form training), one model sees facts only as statements such as `The capital of A is B.`, while the other sees those same facts as statements and in four additional forms, including `Capital of A: B.` and `Name the capital of A: B.`. In the second stage (target-fact training), both models receive identical training on new facts, all written as statements such as `The capital of X is Y.`. In the capitals-only experiment, both models produce `Y` after `The capital of X is` for nearly every evaluated fact. But after `- Capital of X:`, the first model almost never produces `Y`, while the second does so for 97.7% of the evaluated facts. Because their target-fact training is identical, the retrieval difference comes from what they learned before X and Y were introduced.

To understand this difference, we examine the hidden state immediately before the answer, which we call the *context state*. Replacing this state at retrieval time can reverse the result: a state from a successful request can make a failing request succeed, while a state from the failing request can make a successful one fail. A context state from one fact can also help retrieve another fact, even from a different factual relation. This suggests that part of its effect on retrieval can be reused across facts.

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.

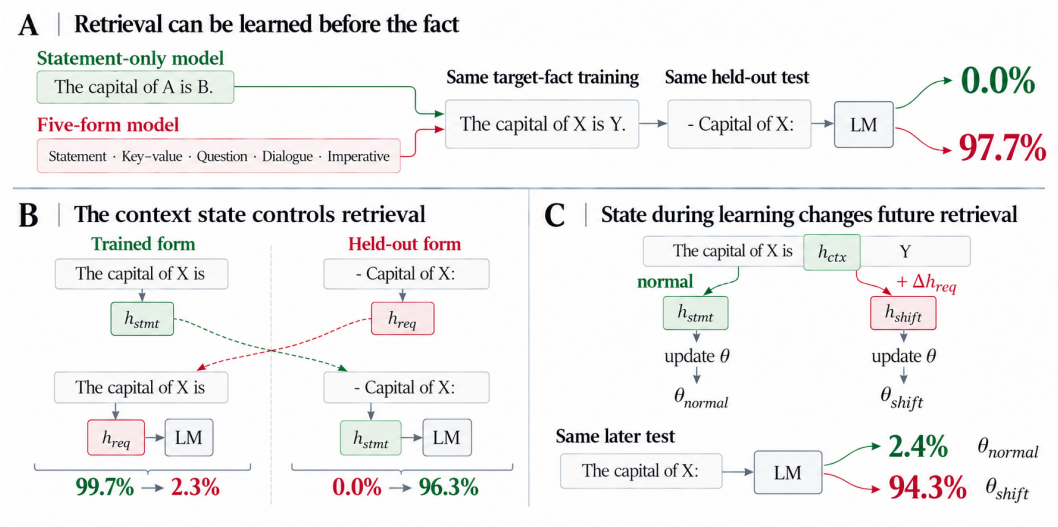


Figure 1: **Learning a fact and learning how to retrieve it are separate, and the context state links them. (A)** Identical target-fact training after different stage-one training: held-out retrieval goes from 0.0% to 97.7%. **(B)** Swapping the context state between a trained and a held-out form removes or restores retrieval. **(C)** Shifting the context state during training changes retrieval through a never-used form afterward, with no intervention at test time.

We next ask whether this state also connects earlier request-form training to later fact learning. For the same new fact, the declarative and list request forms produce more similar context states in the model trained on five forms in stage one than in the model trained only on statements at that stage. To test whether the state present during learning matters, we train a model on the same factual sentences as before but shift its context state toward one produced by a different request form. When training ends, we stop making the shift and test normally. The facts can now be retrieved through a request form that previously failed, without intervention at test time. Thus, how a fact can later be retrieved is shaped in part by the context state present while it was learned.

Our contributions are:

- We separate learning a fact from learning how to retrieve it.
- We identify the *context state* as the link between what the model learns from earlier request forms and how new facts are later retrieved, and show that it can control retrieval across facts and relations.
- We show that changing the context state during factual fine-tuning changes how the learned fact is later retrieved.

2 Related Work

Format experience and knowledge use. Jiang et al. (2024) show that instruction tuning before continued pretraining improves later extraction of knowledge from new documents. Allen-Zhu & Li (2024) show that request-form data mixed into pretraining makes facts about held-out entities extractable. Zhao et al. (2025) map which fine-tuning data formats inject knowledge, and Jiang et al. (2025) separate format alignment from knowledge learning. Multitask prompted training (Sanh et al., 2022; Wei et al., 2022) shows that diverse format experience generalizes to held-out formats. Fine-tuning on new facts is known to generalize narrowly and to be sensitive to presentation (Berglund et al., 2023; Gekhman et al., 2024; Zhang et al., 2024), and facts learned out of context are used only under some prompts (Krasheninnikov et al., 2023; Treutlein et al., 2024), and presentation diversity is the standard remedy (Yang et al., 2025). We hold the later target-fact training fixed, make the earlier-training facts disjoint, wrong, or from another relation, and identify what transfers as a context state that controls retrieval.

Table 1: **Request forms.** The five forms used in stage-one training, and the forms used to evaluate retrieval of the target facts. No evaluation form other than the declarative training form appears in either training stage of the two-stage experiment.

Stage-one training form	**Frame**
Declarative	`The capital of {c} is {y}.`
Key–value	`Capital of {c}: {y}.`
Question	`Q: What is the capital of {c}? A: {y}.`
Dialogue	`User: What is the capital of {c}?\nAssistant: {y}.`
Imperative	`Name the capital of {c}: {y}.`
Evaluation form	**Frame**
Declarative (training form)	`The capital of {c} is`
Possessive	`{c}'s capital is`
Capital-city	`The capital city of {c} is`
Knowledge	`I know that the capital of {c} is`
Relative-clause	`The city that is the capital of {c} is`
Short key–value	`{c} capital:`
List	`- Capital of {c}:`
Colon-ended declarative	`The capital of {c}:`

Superficial alignment. Zhou et al. (2023) and Lin et al. (2024) argue that alignment teaches format over knowledge already present; Kung & Peng (2023) and Min et al. (2022) show that instruction and demonstration data can be exploited for format with little regard to correctness. We agree that the format component is separable and content-free, and show what it is: a context state that controls retrieval, which fact training produces only for the form the facts were trained in.

Where facts live and how they are read. Facts have been located in MLP key–value memories (Geva et al., 2021; Meng et al., 2022), and factual recall has been traced as enrichment at the subject, propagation of the relation to the last position, and extraction there (Geva et al., 2023; Hernandez et al., 2024); task identity can be carried by activation vectors (Todd et al., 2024; Hendel et al., 2023); mean-difference directions steer behavior, including instruction following (Turner et al., 2023; Rimsky et al., 2024; Li et al., 2023; Zou et al., 2023; Liu et al., 2024; Stolfo et al., 2025); surface form competes with content at the output (Holtzman et al., 2021). Learned prefixes, prompts, and interventions on hidden states adapt a model without changing its weights (Li & Liang, 2021; Lester et al., 2021; Wu et al., 2024), and steering or ablating directions during fine-tuning changes what a model learns (Chen et al., 2025; Casademunt et al., 2025). The reversal curse (Berglund et al., 2024) shows that the form of training constrains the direction of retrieval, and training on reversed text lifts it (Golovneva et al., 2024). We show that it constrains the *context* of retrieval, that the constraint is a state at the answer position, and that the direction which lifts it is the difference between the two contexts' answer-position states, obtained from prompts with a placeholder in place of the relation word.

Acquisition versus use. The gap between acquired and usable knowledge (Kazemnejad et al., 2023), format-specific evaluation competence (Bunn et al., 2025), prompt and paraphrase sensitivity (Sclar et al., 2024; Lu et al., 2022; Mizrahi et al., 2024; Jiang et al., 2020; Elazar et al., 2021), and curriculum and warm-start effects (Bengio et al., 2009; Ash & Adams, 2020; Papadimitriou & Jurafsky, 2023; Hu et al., 2025) describe the phenomenon we isolate and explain.

3 Learning a Fact and Learning How to Retrieve It Can Be Separated

Data and model. We study several factual relations, including capitals, currencies, populations, organization founders, and birth years, across Pythia-410M and Pythia-1.4B (Biderman et al., 2023), Qwen2.5-1.5B, and Llama-3.2-1B. Unless stated otherwise, detailed analyses use Pythia-410M and fictional country–capital pairs, with disjoint sets of 1,000 earlier-training facts and 1,000 target facts.

Table 2: **The same facts are learned but differ sharply in how they can be retrieved.** The five relations are learned together in one training run. Each relation is evaluated on its 200 target facts. For every relation, both histories receive identical target-fact training in the declarative training form. Values are exact-generation accuracy. The statement-only values for birth years (0.344 and 0.522) are higher than for the other relations: year answers come more readily after a colon.

	Declarative form		List request form		Colon-ended declarative form	
Relation	**Statement-only**	**Five-form**	**Statement-only**	**Five-form**	**Statement-only**	**Five-form**
Capital	0.983	0.999	0.002	**0.970**	0.200	**0.996**
Currency	0.995	0.999	0.004	**0.989**	0.388	**0.999**
Population	0.991	0.986	0.038	**0.991**	0.276	**0.994**
Founder	0.992	0.992	0.001	**0.989**	0.299	**0.998**
Birth year	0.987	0.986	0.344	**0.982**	0.522	**0.989**

3.1 Two-stage training

The two training stages separate learning the new facts from earlier training on request forms (Figure 1). In the first stage, one model sees each fact in five request forms (the *five-form model*), while another sees the same facts only as statements (the *statement-only model*). In the second stage, both models receive identical training on the target facts, all presented as statements. None of these target facts appears in the first stage, and no evaluation form in Table 1 other than the declarative training form appears in either stage. Because the second-stage training is identical, any difference in how the new facts are later retrieved must come from the first stage.

Evaluation. We evaluate the same fixed set of 300 from 1,000 target facts on which both models are trained in the second stage, under the eight evaluation forms in Table 1. We measure whether the correct capital is ranked first among all 2,000 candidate capitals and, where appropriate, whether the model generates the capital exactly. Unless stated otherwise, results are averaged over eight paired training seeds. The retrieval experiments report the list and colon-ended declarative forms; Figure 2a reports all eight. Full training and statistical details are provided in Appendix A.

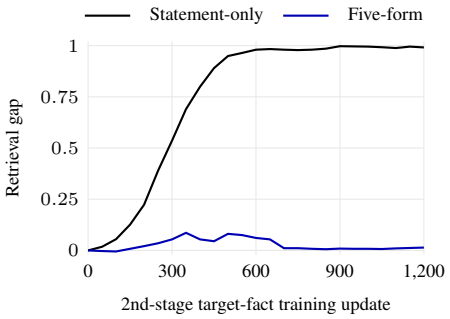
3.2 Learning a Fact Does Not Teach Retrieval Through New Request Forms

Table 2 reports exact-generation accuracy of the two models when five factual relations are trained together. For capital facts, the two models perform similarly in the declarative form used for target-fact training (0.983 and 0.999), yet differ on the list form (0.002 and 0.970) and the colon-ended declarative form (0.200 and 0.996). Neither of these evaluation forms appears in either training stage. The same broad contrast holds for currencies, populations, founders, and birth years: both models answer the target facts in the training form, while earlier request-form training strongly affects retrieval in the other forms.

We use capital facts for a full breakdown across the eight evaluation forms in Table 1. Both models retrieve the target facts almost perfectly from the declarative training form, with exact-generation accuracy of 0.995 and 0.997 (Figure 2a). Their behavior differs sharply on other request forms. The statement-only model fails on the list, short key–value, possessive, and colon-ended declarative forms, while the model with earlier request-form training retrieves the same facts through the list and colon-ended declarative forms with high accuracy. For example, retrieval through the list form, which neither training stage uses, increases from 0.000 to 0.977, despite identical target-fact training. The two forms with no token bigram in common with any earlier-training form, the possessive and the short key–value form, improve little (0.165 and 0.227), and one declarative paraphrase, the relative clause, drops (0.895 to 0.801), in line with the finding in Section 3.4 that retrieval depends on how the request ends.

The retrieval difference is not explained by better learning of the target facts in the training form. The two models perform nearly identically on the declarative form, and continuous measures of learning also show no detectable difference (Appendix K): the first-token negative log-likelihood of the correct answer differs by -0.043 nats and its margin over the strongest incorrect answer by $+0.24$, with

Context	Statement-only	Five-form
Declarative	0.995	0.997
Possessive	0.039	0.165
Capital-city	0.885	0.885
Knowledge	0.987	0.998
Relative-clause	0.895	0.801
Short key–value	0.000	0.227
List	0.000	0.977
Colon-ended declarative	0.024	0.993



(a) Same facts, different retrieval **(b) The retrieval gap emerges during fact learning**

Figure 2: **The retrieval difference extends across request forms and emerges during fact learning. (a)** Both models retrieve the target facts almost perfectly from the declarative training form, but differ sharply across many other request forms. **(b)** During target-fact training, the gap between declarative and list retrieval (the difference in pool top-1 accuracy) grows in the statement-only model, while remaining near zero after earlier five-form training. Both models receive identical target-fact training. Panel (a) reports exact generation; panel (b) reports pool top-1.

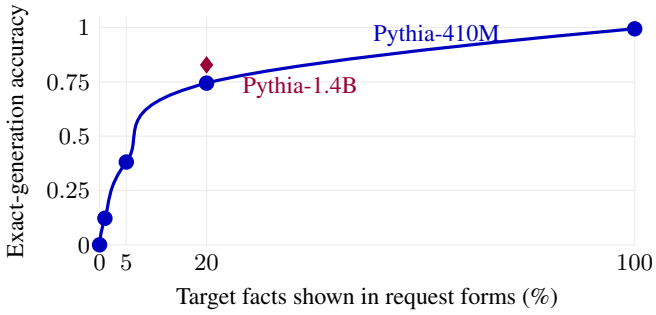


Figure 3: **Retrieval improves with request-form exposure.** Exact-generation accuracy on the list form, for target facts written only as statements; the Pythia-1.4B point is at 20%.

intervals that include zero in both cases. Thus, we find no detectable difference between the models on our measures of target-fact learning, while they differ sharply in retrieving those facts through other request forms.

This central separation—similar performance in the declarative training form but sharply different retrieval in other forms—also appears across model families, factual relations, entity types, answers whose first token is shared with other capitals, LoRA fine-tuning, and independently regenerated fact worlds. The joint five-relation result also replicates with Pythia-1.4B (Appendices F and E).

Figure 2b further shows how the retrieval gap develops during second-stage training. Without earlier request-form training, the model learns the target facts while the retrieval gap between the declarative and list forms grows toward one. After earlier request-form training, the gap remains near zero as the facts are learned. Thus, learning the target facts alone does not teach retrieval through a new request form. Longer second-stage training does not resolve the statement-only model's list-form failure. Extending training from 1,200 to 4,800 updates leaves list-form exact-generation accuracy at 0.000 in every seed. The result is the same with a learning rate five times lower, when training continues until declarative-form accuracy is comparable to the standard run (Appendix C).

Table 3: **Earlier request-form training transfers to new facts.** All models receive identical declarative training on the target capital facts. Mean exact-generation accuracy across the two request forms rises substantially after earlier request-form training, even when the earlier answers are incorrect or come from a different factual relation. In the incorrect-answer condition, every country is paired with another country's capital. The incorrect-answer corpus uses the key–value form; the currency corpus uses key–value and declarative lines; neither is token-matched to the five-form corpus.

Earlier training	**Mean**	**Request form**	
		List	**Colon-ended declarative**
Statements only	**0.012**	0.000	0.024
Correct unrelated capitals	**0.985**	0.977	0.993
Incorrect capital answers	**0.681**	0.639	0.724
Currencies	**0.944**	0.900	0.987

3.3 Earlier Request-Form Training Transfers to New Facts

Earlier training on request forms can later support retrieval of facts that were never shown in those forms. This transfer does not require the same answers or even the same factual relation. We run the earlier training using unrelated capital facts with incorrect answers or using currency facts, then give all models identical declarative training on new capital facts. Both conditions substantially improve later retrieval of the new capitals through several request forms (Table 3).

Request-form training has the same effect whether it uses earlier facts or the target facts themselves. When the target facts are shown in some request forms during training, retrieval also improves in other request forms in which those facts never appeared. The improvement is nearly the same as when the request-form training used different facts earlier. Thus, what the model learns from a request form transfers across facts.

A small amount of request-form exposure is enough to improve retrieval of other facts (Figure 3). Retrieval increases monotonically with exposure, and the same effect is visible in Pythia-1.4B at 20% exposure. At 5%, declarative paraphrases yield only 0.003, compared with 0.381 for request-form exposure (Appendix D).

3.4 Retrieval Depends on the Request Near the Answer

Retrieval is sensitive to the local form of the request near the answer. Across capitals, currencies, populations, organization founders, and birth years, changing only the final words of a request can strongly change whether the fact is retrieved. For example, in the statement-only model, changing a list request from a colon ending to an "is" ending raises exact-generation accuracy from 0.002 to 0.814 for capitals. The same pattern appears across the other relations, and across sixteen request forms fixed in advance, retrieval rises with how many of the training line's final words recur before the answer: 0.02, 0.42, 0.59, and 0.97 for one to four, and 0.02, 0.72, 0.92, and 0.97 among the forms that also keep the relation word (Appendix G).

By contrast, adding a 30-token unrelated passage before the request does not change which request forms can retrieve the facts. We prepend the same unrelated 30-token passage to every request. In the statement-only model, the list and colon-ended declarative forms still fail, while forms that already retrieve the facts continue to do so. After earlier request-form training, the same request forms remain successful despite the added preamble (Appendix B).

Retrieval therefore depends mainly on the final words of the request form, rather than on unrelated text that appears earlier. This points to the hidden state immediately before the answer is produced as a natural place to look for what controls retrieval.

Table 4: **The same direction supports retrieval across request forms and factual domains.** (a) Exact-generation accuracy before and after adding the direction to the context state in the capital task. Each form's direction is scaled to that form's restoring-vector norm. (b) Directions derived from capitals and birth years transfer almost unchanged to the other factual domain, each built in that domain's own model.

(a) Across request forms

Request form	No intervention	+ Direction
List	0.000	**0.921**
Colon-ended declarative	0.024	**0.983**

(b) Across factual domains

Target	Same-domain	Other-domain
Capitals	0.921	0.910
Birth years	0.971	0.969

4 A Reusable Context State Controls Retrieval

Changing the context state alone can turn retrieval on or off; the state that does so transfers across facts and relations; and much of its effect is a single direction.

4.1 Changing the Context State Changes Retrieval

Changing the context state alone can enable or prevent retrieval. Holding the list request form fixed, replacing its context state with the context state produced by the declarative training form raises exact-generation accuracy from 0.000 to 0.963, alongside the 0.948 reached by rewriting the request to end in "is" (Appendix H). A norm-matched random state restores nothing, and replacing the state at the entity token instead of the context state has no effect. The intervention also works in the opposite direction: holding the declarative request form fixed and replacing its context state with the list-form context state reduces accuracy from 0.997 to 0.023. Thus, changing only the context state can either enable or prevent retrieval of an already learned fact.

4.2 The Context State Transfers Across Facts and Relations

The context state that restores retrieval does not have to come from the same fact. A context state produced while asking about one fact can restore retrieval of another fact almost as well as a context state produced from that fact itself. This holds across Pythia-410M, Pythia-1.4B, and Qwen2.5-1.5B, and also when the answers contain two tokens.

The context state can also come from a different factual relation. In Pythia-410M at layer 4, exact-generation accuracy is 0.963 with a context state from the same fact and 0.951 with one from a different fact. Context states from currency and population facts perform almost as well, reaching 0.946 and 0.928.

This effect is not simply due to copying the final token of the request form. At layer 0, replacing the context state is equivalent to replacing that token, but in later layers the useful information persists only when the state carries an entity in the Pythia models. The two components separate by depth: at layer 4 an entity-free state recovers 0.734 of the 0.963, so most of the effect there is the final token, while by layer 6 the entity-free state recovers 0.02 and states carrying an entity still recover 0.70–0.78. Pythia-1.4B shows the same pattern, and the reverse intervention in Qwen2.5-1.5B reduces declarative retrieval from 0.999 to 0.550. Thus, the part of the context state that supports retrieval is not tied to the particular fact or relation that produced it.

4.3 A Fact-Independent Direction Supports Retrieval

Replacing the entire context state is not necessary to restore retrieval. Much of the effect is captured by a single direction: the difference between context states of prompts ending in "is" and prompts ending in a colon, built with a placeholder in place of the relation word (`The value of {c} is` against `The value of {c}:`) so that it carries nothing about capitals. Adding this direction to the context state of a failing list form raises exact-generation accuracy from 0.000 to 0.921 (Appendix J). Removing the same direction from the restoring vector reduces accuracy to 0.010, while removing



Figure 4: **Earlier request-form training keeps request-form context states aligned with the declarative one.** At each layer, centered cosine similarity across target facts between the declarative context state and the context state of the list form; layer 24 is taken after the final layer norm. The two training conditions begin alike and separate in later layers in both models.

other directions costs in proportion to their overlap with it. A learned linear map adds little beyond it (0.949 against 0.921). Thus, a single direction accounts for much of the context state's effect on retrieval.

The same effect holds across model families, factual domains, shared-first-token answers, and training methods (Appendix F). In Qwen2.5-1.5B, list retrieval is already partly supported before any intervention, with accuracy 0.603. Adding the direction raises accuracy to 0.993. Removing the direction from the full restoring vector still leaves accuracy at 0.944, suggesting that Qwen already has other components that support this request form. In the Pythia models, where the list form initially fails, removing this direction removes most of the restoring effect.

The same direction also supports several request forms and transfers across factual domains (Table 4). For capitals, adding it raises list retrieval from 0.000 to 0.921 and colon-ended declarative retrieval from 0.024 to 0.983. A direction derived from capital facts works almost as well for birth-year facts, and vice versa. Thus, the direction is not specific to one request form or factual domain.

5 Context States Link Facts to Future Retrieval During Learning

The context state links earlier request-form training to how new facts are later retrieved. Earlier request-form training changes the context states produced by other request forms, and changing the context state during factual fine-tuning changes which request forms can retrieve the new facts later.

5.1 Request Forms Produce Aligned Context States

Earlier request-form training keeps other request forms aligned with the form used to teach the new facts. We compare the context state produced by the list form with the context state produced by the declarative training form. Without earlier request-form training, these states become increasingly different in later layers (Figure 4). With earlier request-form training, they remain much more similar. The same pattern appears in both Pythia-410M and Pythia-1.4B.

The difference is small at the first layer. The context states separate as they pass through the model, showing that earlier request-form training changes how those forms are transformed internally before the answer is produced.

5.2 Changing the Context State During Learning Changes Future Retrieval

During factual fine-tuning, each new fact is still shown as a declarative sentence, such as "The capital of X is Y." The sentence itself is unchanged. Instead, the context state is moved toward the state produced by another request form, such as "Capital of X:". This change is computed on facts other

Table 5: **Changing the context state during learning changes future retrieval.** The factual training examples are identical across conditions; only the context state present while each fact is learned is changed. Values are exact-generation accuracy on the colon-ended declarative form, whose direction was never used in the shift; details are in Appendix I.

Model	Facts	No intervention	Random shift	Targeted shift
Pythia-410M	Capitals	0.024	0.100	**0.943**
Pythia-1.4B	Capitals	0.266	0.332	**0.904**
Qwen2.5-1.5B	Capitals	0.482	0.722	**0.993**
Pythia-410M	Birth years	0.246	0.445	**0.983**

than those evaluated, so X and Y have never appeared together in that request form. After training, the context state is left untouched and the model is tested normally. If "The capital of X:" can now retrieve Y, the difference must come from the context state that was present while X and Y were learned.

This change produces large differences in later retrieval across models and factual domains (Table 5). In Pythia-410M, accuracy rises from 0.024 to 0.943; in Pythia-1.4B, from 0.266 to 0.904; in Qwen2.5-1.5B, from 0.482 to 0.993; and for birth-year facts, from 0.246 to 0.983. The improvement appears on a request form whose context state was not used to construct the change. Thus, changing the context state while a fact is learned can change how that fact is retrieved later.

The particular change to the context state matters. A random change is much less effective. In Qwen2.5-1.5B, where a random change also improves retrieval, changing the state toward another request form improves accuracy by an additional 0.27. Changing the context state is also more effective than showing a small number of target facts in the new request forms. Writing 5% of the target facts in the request forms gives 0.72 on this form; the shift gives 0.94 without writing any target fact in a request form.

6 Implications and Limitations

Implications. Factual fine-tuning changes both what a model learns and how that knowledge can later be retrieved. These two effects can be separated: identical training on the same new facts can lead to different retrieval depending on earlier request-form training. The context state links the two. After a fact has been learned, changing the context state can enable or prevent its retrieval. During factual fine-tuning, changing the context state changes how the fact is retrieved later.

These results reveal a new way to control factual fine-tuning. The facts themselves can be learned in one request form, while retrieval through other request forms can be learned separately and reused across facts. In the experiments, even a small amount of request-form training improves retrieval of facts that were never shown in those forms. More importantly, changing the context state during factual fine-tuning changes later retrieval without requiring the factual examples to be rewritten in additional request forms. Retrieval can therefore be controlled separately from the factual content being learned.

The results also change how retrieval failures should be interpreted. If a model fails to produce a fact through a particular request form, that does not necessarily mean the fact was not learned. In these experiments, the fact can already be present and retrievable through one request form while failing through another, and changing only the context state can recover it. Evaluating factual fine-tuning therefore requires separating whether the fact was learned from whether the model can retrieve it.

Limitations. The results have three main limits. First, the experiments use controlled factual training, where fact learning and request-form training can be separated precisely; learning facts from natural text may involve additional mechanisms. Second, the context state explains the retrieval failures studied here, but it does not explain every failure to retrieve factual knowledge. Third, the effect depends on what the pretrained model already knows about a request form. Larger models, different pretraining, and instruction tuning may therefore change how strongly request-form training affects later retrieval.

AI USE STATEMENT

Generative AI tools were used to assist with experiment planning, code review, analysis organization, and manuscript drafting and editing. The authors ran and verified the experiments and take responsibility for all claims and artifacts.

REPRODUCIBILITY STATEMENT

Code, prompts, and per-cell result files are available at an anonymous repository[1]. All primary comparisons use paired seeds 0–7 with the seed as the inferential unit. Corpus builders, literal prompts, per-seed outputs, patching scripts, and analysis scripts are included in the supplementary artifacts; every reported statistic can be recomputed from them. Seeds, batch sizes, generation length (6 greedy tokens), literal templates, and search grids are given in the supplementary code.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1067–1077. PMLR, 2024. URL https://proceedings.mlr.press/v235/allen-zhu24a.html.

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3884–3894, 2020.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009. doi: 10.1145/1553374.1553380.

Lukas Berglund, Asa Cooper Stickland, Mikita Balesni, Max Kaufmann, Meg Tong, Tomasz Korbak, Daniel Kokotajlo, and Owain Evans. Taken out of context: On measuring situational awareness in LLMs. *arXiv preprint arXiv:2309.00667*, 2023.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *International Conference on Learning Representations*, 2024.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023.

Alec Bunn, Sarah Wiegreffe, and Ben Bogin. Fine-tune on the format first: Improving multiple-choice evaluation for intermediate LLM checkpoints. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM2)*, pp. 511–521, Vienna, Austria and virtual meeting, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.gem-1.46/.

Helena Casademunt, Caden Juang, Adam Karvonen, Samuel Marks, Senthooran Rajamanoharan, and Neel Nanda. Steering out-of-distribution generalization with concept ablation fine-tuning. *arXiv preprint arXiv:2507.16795*, 2025.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*, 2025.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021. doi: 10.1162/tacl_a_00410.

[1] https://anonymous.4open.science/r/learning-retrieval

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7765–7784. Association for Computational Linguistics, 2024.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.446.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.751.

Olga Golovneva, Zeyuan Allen-Zhu, Jason Weston, and Sainbayar Sukhbaatar. Reverse training to nurse the reversal curse. *arXiv preprint arXiv:2403.13799*, 2024.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.624.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. In *International Conference on Learning Representations*, 2024.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.564.

Michael Y. Hu, Jackson Petty, Chuan Shi, William Merrill, and Tal Linzen. Between circuits and Chomsky: Pre-pretraining on formal languages imparts linguistic biases. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9691–9709. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.478.

Jinhao Jiang, Junyi Li, Wayne Xin Zhao, Yang Song, Tao Zhang, and Ji-Rong Wen. Mix-CPT: A domain adaptation framework via decoupling knowledge learning and format alignment. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=h1XoHOd19I.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020. doi: 10.1162/tacl_a_00324.

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Victoria Lin, Wen-tau Yih, and Srinivasan Iyer. Instruction-tuned language models are better knowledge learners. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5421–5434. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.296. URL https://aclanthology.org/2024.acl-long.296/.

Amirhossein Kazemnejad, Mehdi Rezagholizadeh, Prasanna Parthasarathi, and Sarath Chandar. Measuring the knowledge acquisition-utilization gap in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 4305–4319. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.285. URL https://aclanthology.org/2023.findings-emnlp.285/.

Dmitrii Krasheninnikov, Egor Krasheninnikov, Bruno Mlodozeniec, Tegan Maharaj, and David Krueger. Implicit meta-learning may lead language models to trust more reliable sources. *arXiv preprint arXiv:2310.15047*, 2023.

Po-Nien Kung and Nanyun Peng. Do models really learn to follow instructions? an empirical study of instruction tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1317–1328. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-short.113.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

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*, volume 36, pp. 41451–41530, 2023.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. In *International Conference on Learning Representations*, 2024.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32287–32307. PMLR, 2024.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.556.

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, pp. 17359–17372, 2022.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.759.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681.

Isabel Papadimitriou and Dan Jurafsky. Injecting structural hints: Using language models to study inductive biases in language learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8402–8413. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.563.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.828.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala

Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, 2024.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. In *International Conference on Learning Representations*, 2025.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations*, 2024.

Johannes Treutlein, Dami Choi, Jan Betley, Samuel Marks, Cem Anil, Roger Grosse, and Owain Evans. Connecting the dots: LLMs can infer and verbalize latent structure from disparate training data. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. In *International Conference on Learning Representations*, 2025.

Xiao Zhang, Miao Li, and Ji Wu. Co-occurrence is not factual association in language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Eric Zhao, Pranjal Awasthi, and Nika Haghtalab. From style to facts: Mapping the boundaries of knowledge injection with finetuning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3335.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivas Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pp. 55006–55021, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Exact Protocol

Fact world. We generate 2,000 synthetic country–capital pairs with none a real country–capital pair; A is the first 1,000 and B the next 1,000, disjoint as strings and as first tokens for capitals. The 300 evaluation facts, `evA`, are $A[0{:}300]$, fixed across seeds and conditions. The 2,000-capital candidate pool is fixed. Birth years are single-token years in 1000–2099; the five-relation world has 400 entities per relation, with organization names built as a word plus a type noun; shared-first-token

capitals are vocabulary stems with a suffix, exactly two tokens long, with first tokens deliberately not unique. Every capital used in the main experiments has a unique first token in the pool; the shared-first-token world removes this property.

Training. EleutherAI/pythia-410m, fp32; AdamW $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01; learning rate 5×10^{-5} constant; micro-batch 2, accumulation 4; sequence length 256; gradient clip 1.0; 1,200 updates per phase unless stated; full-sequence causal LM loss with training lines interleaved into Wikitext at sentence granularity about seven sentences apart in the target stage; fresh AdamW at the phase boundary. The five-form earlier-training corpus writes each fact in five request forms: `The capital of {c} is {y}.`, `Capital of {c}: {y}.`, `Q: What is the capital of {c}? A: {y}.`, `User: What is the capital of {c}?\nAssistant: {y}.`, and `Name the capital of {c}: {y}.`; the statement-only earlier-training corpus writes the same facts as statements only and is token-matched to it. Within a seed the target-phase corpus tensor is byte-identical across histories. Each target fact appears in five lines per pass and the stage covers about 2.2 passes, so each fact is seen about eleven times. Pythia-1.4B, Qwen2.5-1.5B, and Llama-3.2-1B use the same recipe in fp32, with the capitals rebuilt under each model's tokenizer. LoRA: rank 64, $\alpha = 128$, learning rate 3×10^{-4}, 2,400 updates, attention and MLP projections, target stage only.

Readouts. Pool top-1 ranks the correct capital first among the 2,000 candidates at the first answer token. Exact generation requires the greedy continuation to begin with the capital. A request form is held out for a condition if it appears in neither training stage of that condition nor of its control.

B Requests With a Preamble

The declarative, list, "is"-ended list, and colon-ended declarative forms are scored with a fixed thirty-token passage unrelated to the facts placed before it. Exact generation, eight-seed means, Pythia-410M, 300 target facts. A training line is the declarative form, the capital, and a period.

	no preamble		with preamble	
request form	statement-only	five-form	statement-only	five-form
declarative	0.995	0.997	1.000	0.999
list	0.000	0.977	0.000	0.995
`- Capital of {c} is`	0.948	0.985	0.972	0.997
`The capital of {c}:`	0.024	0.993	0.011	0.998

C More Target-Fact Training

Statement-only history, Pythia-410M, eight seeds, exact generation. X4800: 4,800 updates at the standard rate. XLR: learning rate 10^{-5}, run until declarative generation matches the standard run's final value (-0.01), checked every 200 updates (at most 9,600), which took 1,000–6,800 updates.

form	standard (1,200 updates)	X4800	XLR
declarative	0.995	0.997	0.988
list	0.000	0.000	0.000
`- Capital of {c} is`	0.948	0.974	0.699
`The capital of {c}:`	0.024	0.045	0.009

D Request-Form Exposure

A fraction of the target facts, drawn from outside the 300 evaluation facts, is written under the five request forms and the rest as statements, with statement-only earlier training; evaluation uses the 300 target facts written only as statements; at 100% the token-matched stage writes 768 of the 1,000 target facts, including all 300 evaluation facts, in the five forms, so both columns remain measurements on unwritten forms. The declarative-paraphrase control writes the possessive and capital-city forms

among its paraphrases; only the list and colon-ended declarative forms are scored. Exact generation, eight-seed means.

target facts in request forms	list	colon-ended declarative
0%	0.000	0.024
1%	0.122	0.508
5%	0.381	0.718
20%	0.744	0.880
100%	0.994	0.998
5%, declarative paraphrases instead	0.003	0.062
Pythia-1.4B, 20%	0.828	–

E REPLICATIONS

Independent fact worlds. Two additional worlds were generated from independent generator seeds and run under the unchanged protocol. Earlier five-form training exceeds the statement-only condition on held-out list retrieval (pool top-1) by $+0.974$ and $+0.887$, $8/8$ seeds in each world; eight further seeds on the second world give $+0.972$, $8/8$.

F GENERALITY

The separation, the dependence on the request's final words (keying), and the restore by rewrite or by the restoring vector, in every setting. Contrast columns: exact generation on the held-out list form after identical target-fact training, without and with earlier request-form training on unrelated facts. Keying: the statement-only model on `- {Rel} of {e} is` versus `The {rel} of {e}:`. Restore: the statement-only model's held-out list form, rewritten to end in "is" / with the model's own restoring vector (its mean declarative minus mean list context state) added at the answer position; the direction alone, used in Table 4, gives 0.921 in Pythia-410M. Eight seeds unless marked; Llama-3.2-1B four seeds. The five-relation rows are one training run on 1,000 facts across five relations and three entity types; declarative training-form accuracy is 0.97–1.00 in every row. LoRA: rank 64, target stage only.

	list form	
setting	statement-only	five-form
Pythia-410M, capitals	0.000	0.977
Pythia-1.4B	0.040	0.997
Qwen2.5-1.5B Base	0.603	0.997
Llama-3.2-1B Base	0.048	0.997
Five relations: capital	0.002	0.970
currency (country)	0.004	0.989
population (country)	0.038	0.991
founder (organization)	0.001	0.989
birth year (person)	0.344	0.982
Birth years as target	0.150	0.952
Shared-first-token capitals	0.001	0.982
Targets written by LoRA	0.000	0.836

setting	keying (is / colon)	restore (base → rewrite / restoring vector)
Pythia-410M, capitals	0.948 / 0.024	0.000 → 0.948 / 0.945
Pythia-1.4B	0.994 / 0.266	0.040 → 0.994 / 0.917
Qwen2.5-1.5B Base	0.998 / 0.482	0.603 → 0.998 / 0.993
Llama-3.2-1B Base	0.995 / 0.125	0.048 → 0.995 / 0.971
Birth years as target	0.977 / 0.221	0.150 → 0.977 / 0.973
Shared-first-token capitals	0.941 / 0.025	0.001 → 0.941 / 0.934
Targets written by LoRA	0.708 / 0.010	0.000 → 0.708 / 0.730

The five-relation run on Pythia-1.4B (eight seeds, list form, exact generation; every relation separates in $8/8$ seeds with matched declarative accuracy):

relation	statement-only	five-form
capital	0.119	0.998
currency	0.256	0.999
population	0.248	0.991
founder	0.089	0.996
birth year	0.685	0.996

The five relations learned in one run on Pythia-410M, keying (is / colon) and restore by rewrite:

relation	keying (is / colon)	restore by rewrite
capital	0.814 / 0.200	$0.002 \rightarrow 0.814$
currency	0.730 / 0.388	$0.004 \rightarrow 0.730$
population	0.960 / 0.276	$0.038 \rightarrow 0.960$
founder	0.862 / 0.299	$0.001 \rightarrow 0.862$
birth year	0.987 / 0.522	$0.344 \rightarrow 0.987$

The birth-year statement-only list value (0.344) is higher than for the other relations: year answers come more readily after a colon; the five-form value is unaffected.

G Suffix Bank

Sixteen request forms fixed before scoring, one to four per level of overlap with the training line `The capital of {c} is` (overlap = number of the training line's final tokens that recur immediately before the answer). Exact generation, eight-seed means, statement-only model.

overlap	request form	generation
0	`- Capital of {c}:`	0.000
1	`{c}'s capital is`	0.039
1	`{c}'s capital city is`	0.002
1	`For {c}, the capital is`	0.022
2	`The capital city for {c} is`	0.546
2	`The capital in {c} is`	0.891
2	`The seat of government for {c} is`	0.164
2	`The chief city in {c} is`	0.085
3	`The capital city of {c} is`	0.885
3	`- Capital of {c} is`	0.948
3	`The seat of government of {c} is`	0.359
3	`The main city of {c} is`	0.187
4	`I know that the capital of {c} is`	0.987
4	`The city that is the capital of {c} is`	0.895
4	`Indeed, the capital of {c} is`	0.995
4	`So the capital of {c} is`	0.992

H Patching Details

Target: statement-only model after target-fact training, list form, facts correct under the declarative form by pool top-1. Sites: last country token and answer position. Layers 0–6. Layer ℓ here is the residual stream entering block ℓ; in the direction, shift, and linear-map analyses it is the output of block ℓ. Sources: the same model's declarative answer-position state for the same fact; a norm-matched random vector; the same model's declarative state for a different fact; the cross-relation and entity-free sources are the same model's answer-position states under `The currency of {c} is`, `The population of {c} is`, and `The weather today is` (one state, broadcast to all items). The patched state is held at the prompt's final position on every decoding step; ranking is read from the first forward pass and generation from the held run.

Pythia-410M, full table. Exact generation by layer (0–6) and source, eight-seed means; unpatched 0.000; rewrite 0.948. At layer 0 every non-random source is the final token and equals the rewrite; the layer-0 cells and the rewrite are scored on all 300 facts. Patching at the country token leaves generation at 0.000 at every layer.

source	0	1	2	3	4	5	6
same fact	0.948	0.950	0.963	0.962	0.963	0.947	0.779
different fact	0.948	0.952	0.958	0.959	0.951	0.931	0.701
currency	0.948	0.949	0.961	0.960	0.946	0.934	0.757
population	0.948	0.951	0.959	0.953	0.928	0.920	0.735
no entity	0.948	0.927	0.895	0.834	0.734	0.543	0.018
random	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Pythia-1.4B. Same protocol on the Pythia-1.4B statement-only finals (eight seeds; on the facts correct under the declarative prompt, unpatched generation 0.040; rewrite 0.995). Exact generation by layer and source.

source	0	1	2	3	4	5	6
same fact	0.995	0.996	0.987	0.977	0.962	0.890	0.872
different fact	0.995	0.995	0.985	0.974	0.965	0.895	0.874
currency	0.995	0.995	0.988	0.983	0.971	0.926	0.928
population	0.995	0.995	0.988	0.982	0.972	0.923	0.908
no entity	0.995	0.995	0.948	0.916	0.723	0.448	0.140
random	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Qwen2.5-1.5B. Same protocol on the Qwen2.5-1.5B statement-only finals (eight seeds; unpatched generation 0.603; rewrite 0.998). Every "is"-final source restores the list form at every layer 0–6 (same fact 0.998–0.995, different fact 0.998–0.994, currency 0.998–0.994, population 0.998–0.994, entity-free 0.998–0.989); the norm-matched random state gives 0.000 at every layer. The reverse intervention reduces declarative generation from 0.999 to 0.550; a norm-matched random removal leaves it at 0.999.

Shared-first-token world. Same protocol, on all 300 facts and with the same-fact, different-fact, and random sources, on the statement-only finals of the shared-first-token world (eight seeds; unpatched generation 0.001; rewrite 0.941). Exact generation of both capital tokens by layer and source.

source	0	1	2	3	4	5	6
same fact	0.941	0.941	0.943	0.945	0.932	0.910	0.724
different fact	0.941	0.942	0.938	0.936	0.922	0.892	0.633
random	0.000	0.000	0.000	0.000	0.000	0.000	0.000

I Context-State Intervention During Learning

From the statement-only stage-1 checkpoint, the byte-identical target stream is written for 1,200 updates while $d_f = \bar{h}_4^f - \bar{h}_4^{\text{decl}}$ (the mean layer-4 context state of request form f minus the mean declarative one, on the earlier-training facts; f drawn per line from list, key–value, question, or none, with a seeded RNG) is added at the answer position of every target line at layer 4, moving the state toward the request form. For birth years, d_f is computed on people 600–999 of the birth-year world (target-stream facts outside the evaluation set), in the capital-world stage-one model. Random: norm-matched Gaussian directions with the same draws. Held-out WikiText loss after training (Pythia-410M): targeted shift 5.584, random shift 5.595. Exact generation on `evA`, eight-seed means (birth years: four seeds); the main-text values are the colon-ended declarative form, whose direction was not used in the shift.

model	facts	form	no intervention	random shift	targeted shift
Pythia-410M	capitals	declarative (training)	0.995	0.997	0.995
		colon-ended declarative	0.024	0.100	0.943
Pythia-1.4B	capitals	declarative (training)	0.995	0.997	0.997
		colon-ended declarative	0.266	0.332	0.904
Qwen2.5-1.5B	capitals	declarative (training)	0.999	0.999	0.998
		colon-ended declarative	0.482	0.722	0.993
Pythia-410M	birth years	declarative (training)	0.997	0.992	0.993
		colon-ended declarative	0.246	0.445	0.983

J Context-Direction Vector Details

Grid: $\ell \in \{4, 6, 8, 10, 12\}$, $\beta \in \{0.5, 1.0, 1.5, 2.0\}$. Selection per seed on A[300:600] by exact generation with the vector added at the request's answer position on every decoding step; the selected cell is recorded before `evA` is scored; the selected layer is 4 in every seed. Qwen2.5-1.5B fixes the layer at 4 and selects only β; Pythia-410M and Qwen2.5-1.5B break ties on pool top-1. The Appendix F restoring vectors select layer and β on this grid for every model, keeping the first best cell. The shared-first-token world's grid is offset one layer earlier. Fixed-norm removal: u is projected out of r and the result rescaled to $\|r\|$. The reverse intervention replaces the declarative answer-position state with the same fact's list-prompt state. Linear map: $Mh_4^f(c) \approx h_4^{\text{decl}}(c)$ an affine map fit by ridge regression on the earlier-training facts B[0:600] over the seven non-declarative evaluation forms, λ chosen on B[600:1000], applied at the answer position and held on every decoding step; on `evA`: list 0.949, declarative 0.995; a rank-16 map gives the same values within 0.01. Direction alone: $\|r\|\,\hat{u}$ added at the same site, β chosen on A[300:600] from $\{0.25, 0.5, 1, 1.5, 2, 3\}$ ($\beta = 1$ in every seed). Matched-cosine removals: a direction v with $\cos(v, r) = \cos(u, r)$ is subtracted from r and the result rescaled to $\|r\|$; the mirror of u about r has $\cos(v, u) = 0.603$ and leaves 0.388; ten random v at that cosine have mean $\cos(v, u) = 0.801$ and leave 0.151; u itself leaves 0.010. In Qwen2.5-1.5B the random removal subtracts a norm-matched random direction and rescales to $\|r\|$. The punctuation direction u_ℓ used for the projection test is the difference of answer-position states at layer ℓ between `The value of {c} is` and `The value of {c}:`, averaged over the 300 evaluation country names; built instead from the birth-year world's people, the same direction has cosine 1.000 with it; cosines with that layer's own list vector are 0.895, 0.840, 0.833 at layers 4, 6, 8.

K Statistical Protocol

The training seed is the inferential unit. Core comparisons use eight paired seeds; regenerated worlds are analysed as separate eight-seed comparisons. Exact paired sign-flip tests enumerate all 2^8 sign assignments; with eight same-signed differences the minimum two-sided p is 0.0078125; the $8/8$ counts reported in Appendices E and F are these tests. Confidence intervals use paired seed differences and the t distribution with seven degrees of freedom, untruncated. Interventions applied to other request forms or during training carry a declarative-accuracy check: one that changes declarative accuracy by more than 0.05 in any seed is reported as uninterpretable, never as a null; no such intervention exceeded it.