THE ROLE OF ATTENTION LAYERS AND CONTEXT TOKENS IN ENTITY COPYING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) perform reliably on the task of *entity copying*, in which the model copies the tokens referring to an entity, or *entity tokens*, from the prompt into its outputs to answer a question. Although entity copying is straightforward for most LLMs, existing research does not provide a systematic account of which layers specialize in this fundamental task or how other tokens in the same sequence, termed *context tokens*, influence the model's ability to copy the entity tokens. To address these questions, we conduct experiments on Qwen3-8B using two novel methodologies: *genie-in-a-bottle*, which controls exactly which layers can participate in an entity copying task, and *attention lobotomy*, which cuts off specific tokens' attention to entity tokens without affecting the remaining attention distribution. We find that layers in the second half of the model are both necessary and sufficient for entity copying, whereas those in the first half play little to no role. Moreover, context tokens' joint attention to entity tokens plays a decisive role in successful entity copying, even though these context tokens do not store entity information in their intermediate states unless they satisfy particular semantic properties. Our findings establish the critical role of late layers in entity copying under the decisive guidance of context tokens, calling for future work on how entity information is propagated and consumed in models.

1 INTRODUCTION

Large language models (LLMs) built on the transformer backbone (Vaswani et al., 2017) achieve their impressive performance on natural language generation tasks primarily through the attention mechanism, which allows the current token position to retrieve information from earlier tokens in the sequence. A particularly fundamental task that many models can perform is *entity copying* (Wang et al., 2023; Merullo et al., 2024; Tigges et al., 2024; Feucht et al., 2025; Elhelo & Geva, 2025), in which the model needs to copy the exact tokens that refer to an entity, or *entity tokens*, from the prompt to the output to answer a question about the entity correctly. For example, the model needs to copy "Einstein" from the prompt "Q: My friend is Einstein. What is my friend's name? A:" to produce the output "Einstein". Entity copying is critical to study because it requires the basic model capacity to extract the literal meaning of a token and transmit it to a later token position, an ability required for any task that involves retrieving information from context. Although existing work has examined attention heads sensitive to entity copying (Elhelo & Geva, 2025) and the differences between copying concepts and copying literal words (Feucht et al., 2025), insufficient attention has been paid either to the role of groups of layers at different depths or to that of tokens in the broader context that attend to entity tokens, which we call *context tokens*.

Recent work suggests that, outside the entity copying task, different layers may specialize in entirely different subtasks (Meng et al., 2022; Geva et al., 2023). For example, in tasks that require different layers to collaborate, such as deciding when to insert a line break under a character-count constraint, models have been found to use earlier attention layers to determine the current character's position in the line and later layers to calculate the distance from the character limit, showing specialized layers for each subtask (Gurnee et al., 2025). Similarly, attention to context tokens has been found to contribute meaningfully even when the context tokens themselves do not contain information to be retrieved. For example, the mere presence of extra whitespace tokens can significantly increase the model's ability to recall a particular fact to answer a question (Gekhman et al., 2026), and the verbalization of objects in an image can help models perceive more accurately (Lu et al., 2026). For

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.

the task of entity copying, however, there is still insufficient work on the role of specific layer groups and the impact of specific context tokens.

Existing methodologies are ill suited to studying the roles of layers and context tokens in entity copying. The standard method for attributing roles to layers in entity information retrieval, which injects hidden states from a layer attending to the entity into an uncontaminated prompt (Ghandeharioun et al., 2024; Cheng et al., 2026), falls short of controlling for confounds such as the number of layers available to complete the forward pass after the injection. As a result, we propose our first method, *genie-in-a-bottle*, to capture entity information transmission within a fixed sliding window of activated layers, ensuring that each layer's contribution is carefully controlled. Moreover, existing techniques study context tokens' influence naively by enabling or disabling the model's attention to them (Geva et al., 2023). As such, they cannot disentangle context tokens' contribution of supplying relevant semantic information themselves from their more implicit contributions, such as influencing how strongly the model attends to other tokens. As a remedy, we propose our second method, *attention lobotomy*, to sever selected tokens' attention to the entity token while preserving their complete hidden states, thereby providing a clean way to measure the causal effect of the content of their states on the model's entity copying ability.

Applying the two techniques to Qwen3-8B (Yang et al., 2025), we uncover causal patterns in its entity copying computations associated with specific groups of layers and context tokens. Regarding layer roles, we find that entity copying is almost entirely handled by the last third of attention layers. Entity information can be successfully copied even when the first two-thirds of layers do not contribute, but copying rarely succeeds if some of the late layers stop participating. Regarding context tokens, we find that, even when the model attends to entity tokens directly, the exact words corresponding to those tokens often cannot be recovered if context tokens do not jointly attend to the entity tokens. In some scenarios, the absence of attention from context tokens to entity tokens consistently triggers a chain reaction: the model's attention to the entity also decreases significantly, which in turn causes its outputs to degenerate. Although context tokens can drive the model's allocation of attention to the entity tokens, they themselves do not extract enough entity information to enable entity copying unless they satisfy very specific semantic properties. Specifically, context tokens that are factually correct (for example, "a physicist" rather than "an athlete" for "Einstein") and highly descriptive of the entity (for example, "a physicist" rather than "a human") can sometimes extract enough entity information for use in copying. Together, we find that entity copying activates a highly specialized group of late layers that awaits directions from context tokens to allocate the attention budget.

We summarize our key findings below:

- **Two interpretability methodologies to control causal factors in attention computation.** We propose *attention lobotomy* and *genie-in-a-bottle* to study the contributions of isolated features, such as attention by specific token or layer groups, to entity copying tasks.
- **The specialized role of late layers in entity copying.** We establish the necessary and sufficient role of the last third of layers in Qwen3-8B for entity copying.
- **The guiding role of context tokens in determining the strength of attention to the entity.** We find that context tokens aid entity copying not by storing entity information in their hidden states but by guiding the model to allocate more attention to the correct entity token. They can store entity information if certain semantic properties are satisfied.

2 SETUP

Datasets. To study how the model copies diverse entity tokens across different contexts and cognitive tasks, we collect 100 entities and place them in five prompt templates. We partition the 100 entities into two groups: 50 consist of a single token, such as "Messi" or "Einstein", and the other 50 consist of two tokens, such as "Emily Dickinson". We study both groups to control for any changes in the model's strategy arising from its need to copy more than one token from the context. All entities are placed at the same location within a given template (`direct_fact` (DF), `visitor_register` (VR), `friend` (F), `name_badge` (NB), or `person_in_list` (PIL)), each of which poses a natural-language question whose answer depends on copying the entity tokens (Figure 2). Appendix A provides the complete templates and entity lists.

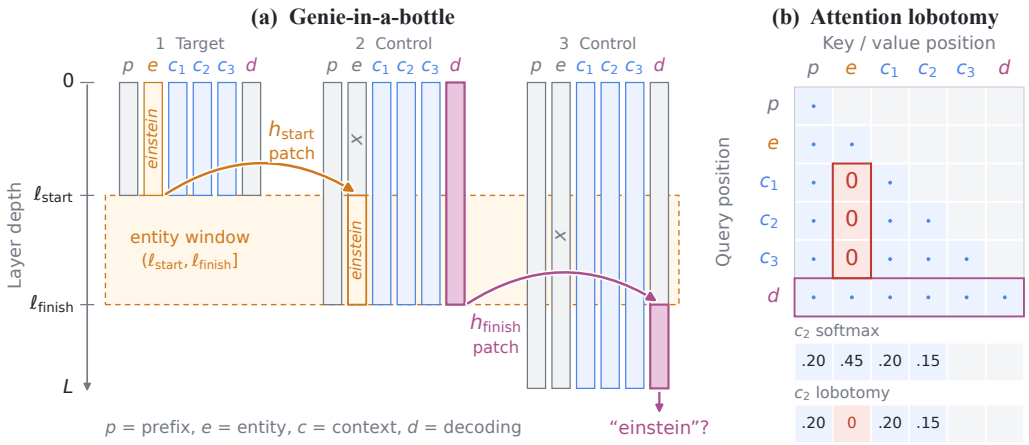


Figure 1: **Two methodologies for studying entity copying.** **(a)** Genie-in-a-bottle confines access to the entity token E to the layer window from ℓ_{start} to ℓ_{finish} by performing three separate forward passes before, within, and after the window. Only the second pass, within the window, enables attention to E. **(b)** Attention lobotomy removes specific attention pathways $x \to y$ to study their effect on entity copying. The effective attention weights for selected destination tokens y are zeroed after softmax.

Model. All experiments use Qwen3-8B-Base (Yang et al., 2025) in completion mode without a chat template to frame entity copying as a next-token prediction task. The model has 36 transformer blocks, each consisting of a full grouped-query attention (GQA) layer (Ainslie et al., 2023) and an MLP layer. Each attention layer has 32 query heads and 8 key/value heads. We use greedy decoding (temperature zero) for reproducible completions, as stochasticity is irrelevant to this study.

Metrics. We judge the success of entity copying for each entity case by whether the correct entity tokens occur in the model's completion of up to 12 tokens. We classify outputs into five categories: `exact_name` copies the exact entity tokens (such as "einstein" for "einstein"); `nonexact_name` recovers tokens that do not exactly match the entity tokens but nevertheless correctly refer to the entity ("albert" for "einstein"); `wrong_name` produces tokens that refer to another entity ("newton" for "einstein"); `not_name` outputs tokens that are coherent but do not include any entity tokens ("i do not know"); and `malformed` produces tokens that are ungrammatical, signaling a disruptive intervention during output generation. For each template, we report both `exact_success_rate`, the rate at which the model produces `exact_name`, and `general_success_rate`, the rate at which the model produces either `exact_name` or `nonexact_name`, and monitor abnormal proportions of `malformed` outputs.

(a) DF

```
Remember this fact: the person's
name is Einstein.
Question: What is the person's
name?
Respond with only the name.
Answer: Einstein
```

(b) PIL

```
Among apple, mouse, Einstein, and
flute, exactly one item is a
person's name.
Respond with only that person's
name.
Answer: Einstein
```

Figure 2: Two of the five prompt templates, shown with the same entity and expected answer.

Methodologies. We introduce two interpretability techniques, illustrated in Figure 1, to study the roles of layer groups and context tokens in entity copying.

We propose *genie-in-a-bottle* to determine whether attention layers within a fixed window from ℓ_{start} to ℓ_{finish} directly extract entity information and, if not, whether these layers contribute indirectly to entity copying through some other mechanism. For a given entity to copy, we prepare two sequences: a target sequence in which the entity tokens are present ("Q: einstein is my friend. who is my friend. A:") and a control sequence in which the entity tokens are replaced by a control token ("Q: x is my friend. who is my friend. A:"). Then, to confine access to the entity tokens to the window from ℓ_{start} to ℓ_{finish}, we run three forward passes sequentially: first, we run the target sequence up to ℓ_{start} and extract the hidden state h_{start} at the entity token position; second, we patch h_{start} into the corresponding token position at ℓ_{start} in the control sequence, run the control sequence up to ℓ_{finish}, and extract the hidden state h_{finish} at the decoding token position; third, in a fresh forward pass of the control sequence, we patch h_{finish} into the same token position at ℓ_{finish} and continue the forward pass and decode the output to assess copying success. Across these three forward passes, the last token position can extract entity information only within the window from ℓ_{start} to ℓ_{finish}. Furthermore, to prevent entity information from leaking into context tokens outside the window, context tokens cannot attend to the entity token position in either the target or the control sequence.

We also propose *attention lobotomy* to determine whether a particular attention pathway between two tokens in the context causally affects entity copying. In particular, we examine whether attention from context tokens to entity tokens matters, and whether the decoding token position needs to attend to context tokens at all. To determine the impact of a given attention pathway $x \to y$ from token x to token y, we set the effective attention weight for y to zero after computing the softmax attention weights from x to all previous tokens. We then inspect the decoded outputs to see whether entity copying is still successful. Zeroing the effective attention weight for y after softmax ensures that the attention weights for all other tokens $t \neq y$ remain unchanged at that operation. Otherwise, any degradation in the decoded outputs can be attributed to the confounding effect of inappropriate attention to tokens $t \neq y$.

3 Which layers are critical for entity copying?

We study the role of layers at different depths in entity copying. In particular, we ask which windows of consecutive layers are necessary or sufficient for extracting entity information from the entity tokens. In addition, for layers that contribute little to extracting entity information, we explore whether some of their computations are still necessary for entity copying to succeed.

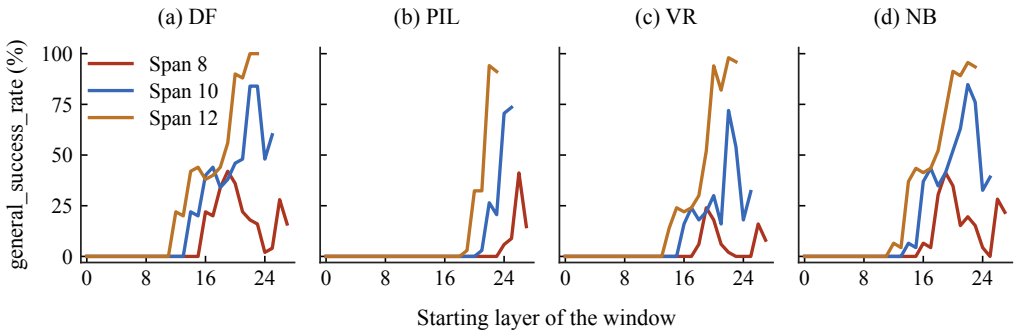


Figure 3: The `general_success_rate` when only a fixed window of attention layers can attend to entity tokens, for different window spans.

Layers in the second half are sufficient for extracting entity information. We apply genie-in-a-bottle to four of the five templates following filtering (see Appendix A), using fixed window sizes of 8, 10, and 12 layers (Figure 3). We use each window's `general_success_rate` as a proxy for how well it extracts entity information. To ensure that entity extraction comes directly from the decoding position's attention to the entity tokens, we zero out all context tokens' attention to the entity tokens. In addition, we remove the decoding position's attention to the entity token position outside the window, where that position contains the control token rather than the target entity token by design. This restriction is motivated by the established observation that the norms of layer outputs increase

with depth (Kimi Team, 2026; Sun et al., 2024), so we posit that layers attending to the control token might interfere with those attending to the target token.

As shown in Figure 3, we find that, with a window size of 8, few windows beginning in the first half produce a nonzero success rate. In contrast, as the window slides past the boundary into the second half, its `general_success_rate` begins to rise. Moreover, with wider windows, we find that success rates consistently increase with layer depth, despite sharp drops at the latest window positions for a window size of 10. This indicates that layers in the second half are sufficient for entity extraction, with later layers making a greater contribution.

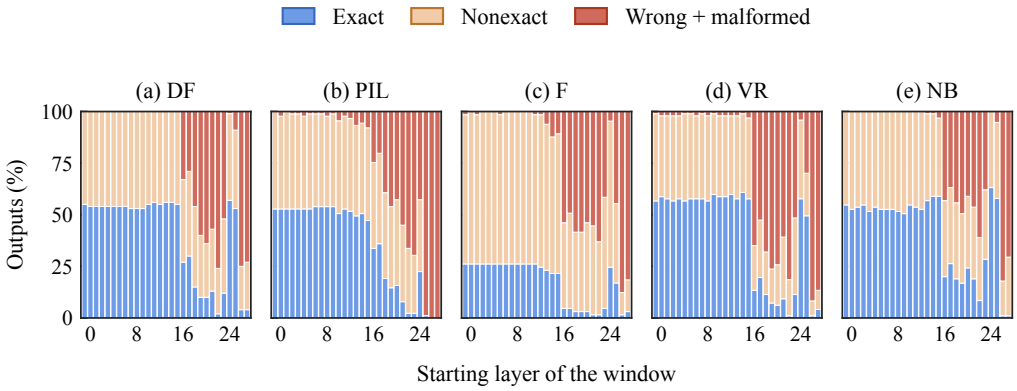


Figure 4: Output distribution after removing attention to entities within a fixed window.

Layers in the second half are also necessary for extraction, with a few deep layers as exceptions. We apply attention lobotomy to the five templates using a window size of 8, zeroing out attention to the entity within each window. Correspondingly, we examine the proportion of outputs classified as `wrong_name` or `malformed` as an indication of whether the window is causally important for entity copying to occur at all. We find that windows in the second half are critical for entity copying, as disabling attention within windows in this region increasingly produces wrong or malformed outputs (Figure 4). In contrast, removing attention within windows in the first half increases the share of either category only minimally. Surprisingly, however, zeroing out attention within a window starting around L24 causes little degradation in four of the five templates. This shows that not all deep layers are useful for entity extraction. Rather, layers in this window are not specialized in extracting entity information.

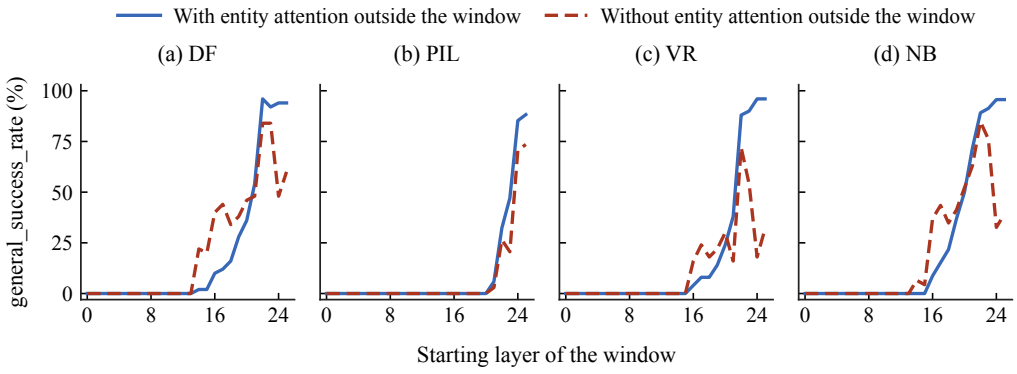


Figure 5: Comparison of `general_success_rate` with attention to the control entity enabled versus disabled outside the designated window.

Although early layers do not participate in entity extraction, their attention to the same position is critical to deep layers' success in extracting entity information. We run genie-in-a-bottle

again on the same four templates with a window size of 10, but instead of removing attention to the entity position outside the window, we allow layers to attend to that position, which contains the confounding control token. As a result, layers outside the window attend to the control token and write their outputs back into the residual stream, competing with information about the entity token. Although we expect degradation across all windows because of this shared confound, we find that `general_success_rate` generally increases across windows (Figure 5). Windows deeper into the second half benefit in particular from early layers attending to the entity position containing the competing token, with higher success rates and no signs of collapse as the windows move deeper. Since the control token is randomly selected for each run and bears no semantic similarity to the target entity token (e.g., "Einstein" versus "Messi"), this suggests that early layers' attention to the same token position is critical to later layers' ability to extract entity information.

4 Can the model copy the exact entity tokens without context tokens attending to them?

The model needs context tokens' joint attention to the entity to copy the literal entity tokens, whereas it can recover the concept of the entity by attending to the entity alone. We perform attention lobotomy by removing the context tokens' attention to the entity token while retaining every other attention pathway. In particular, the decoding position can still attend to both the context tokens themselves to determine the required output format and the entity tokens to copy them one by one. We find that, despite being able to attend to each individual entity token, the decoding position has trouble copying the exact tokens in the prompt (Figure 6(a)). Instead, almost half the time, the model outputs `nonexact_name` responses, such as "George" for "Orwell" (F) and "J. R. R. Tolkien" for "Tolkien" (NB). Therefore, we show that without context tokens' joint attention to the entity, the model often outputs a different set of tokens referring to the same entity, extracting only the concept of the entity rather than its exact tokens. This indicates that context tokens' attention to the same entity is often necessary for copying the literal entity tokens, and that copying entity tokens and extracting the concept of the entity require different computations.

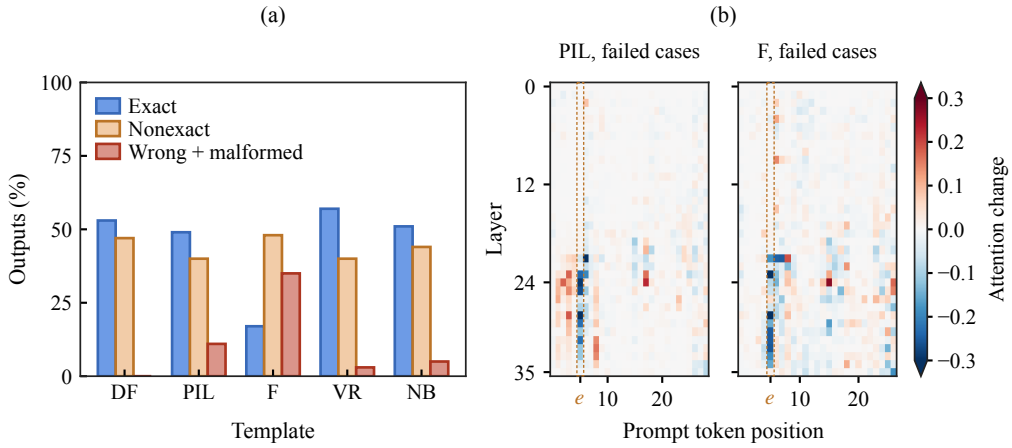


Figure 6: (a) Breakdown of output categories for each template after removing context tokens' attention to entities. (b) For the two templates with the most wrong or malformed outputs, the decoding position's attention to entities decreases sharply after context tokens stop attending to them.

Removing context tokens' attention to the entity can also reduce subsequent positions' attention to the same entity. On two of the five templates where we apply attention lobotomy, F and PIL, we find a particularly high proportion of `wrong_name` and `malformed` outputs, rather than predominantly `exact_name` and `nonexact_name` outputs as in the other three templates (Figure 6(a)). This suggests a different causal mechanism driving the incorrect outputs. At each layer of the decoding position, for each context token it attends to, we plot the change in the mean, across examples, of

the maximum attention score across heads in Figure 6(b), which reveals how peak attention at each layer changes after attention lobotomy on context tokens. We find a large reduction in the late layers' attention to the entity tokens for both templates, in contrast to little or no change in attention for the other three templates. When we restore the decoding position's attention to the entity in each head to its value before attention lobotomy, almost all outputs return to `exact_name` or `nonexact_name` rather than incorrect outputs. This indicates a causal chain for the two templates: removing context tokens' attention to the entity tokens can directly reduce subsequent positions' attention to the same entity, and this reduction in attention in turn produces incorrect outputs.

Restoring context-token attention to entity tokens in just two layers can improve entity copying. Following attention lobotomy on context tokens in F and PIL, we use the same 50 one-token entities per template across conditions and ask whether restoring context tokens' attention to entity tokens in a small group of layers improves subsequent tokens' entity-copying success. To this end, we selectively restore native attention to entity tokens in two layers at context-token positions, while keeping the final prompt position's access unrestricted and generated positions' access blocked. Specifically, we compare early (L5–6), mid (L12–13), and late-mid (L19–20) windows, using zero-based layer indices as in Figure 7(a), to examine how layer depth influences copying recovery. We find that, in early, mid, and late-mid order, `exact_success_rate` rises from the fully blocked baseline of 26% to 98%, 82%, and 42% for F, and from 78% to 86%, 88%, and 90% for PIL (Figure 7(a)). The improvement with each tested window indicates that restoring context tokens' entity access in just two layers can improve copying while direct entity access at the final prompt position remains available. Moreover, recovery with multiple disjoint windows shows that this effect is not restricted to a single tested window.

5 Do context tokens store entity information in their own states?

Given our finding that context tokens' joint attention to entity tokens causally affects subsequent tokens' ability to copy those tokens, an intuitive hypothesis is that context tokens use this attention to store entity information in their own states. At decoding time, the decoding position can therefore retrieve some entity information from context tokens, improving its copying success rate. We therefore design ablations to test this hypothesis against the competing, less committal hypothesis that context tokens' attention to entities merely guides subsequent attention and does not help store entity information in the context.

Despite their causal role in entity copying, context tokens themselves generally do not store entity information. To test the hypothesis that context tokens store entity information in their own states, we apply attention lobotomy to the decoding position's attention to the entity tokens. As a result, the decoding position can attend only to context tokens. If context tokens store entity information in their own states, the model should be able to retrieve that information from them and achieve modest success rates in entity copying. However, we find that almost all outputs under attention lobotomy are `wrong_name` or `malformed` (Figure 7(b)). This indicates that, despite contributing causally to successful entity copying through their attention to entity tokens, context tokens like those in the five templates generally do not store entity information themselves. Instead, they merely guide subsequent tokens to extract entity information effectively.

Description tokens in the context can store entity information only if they are factually correct about the entity. Although context tokens generally do not store entity information, we want to study whether the semantic features of particular token types can predict their ability to do so nonetheless. In particular, under attention lobotomy that removes the decoding position's attention to entities, we append a short indefinite description of each entity's profession after the entity tokens, such as "a physicist" for "Einstein" or "a writer" for "Shakespeare", to study whether the description can store entity information. An indefinite description is one that cannot uniquely identify an entity, such as "a physicist" for either "Einstein" or "Newton". By using an indefinite description in the prompt, we ensure that successful copying of the entity tokens cannot result from the model guessing the entity from the description. We find that, although most outputs remain incorrect, as they do without the description, the absolute number of successful outputs increases substantially (Figure 8). This indicates that, under specific circumstances, indefinite descriptions do store entity information.

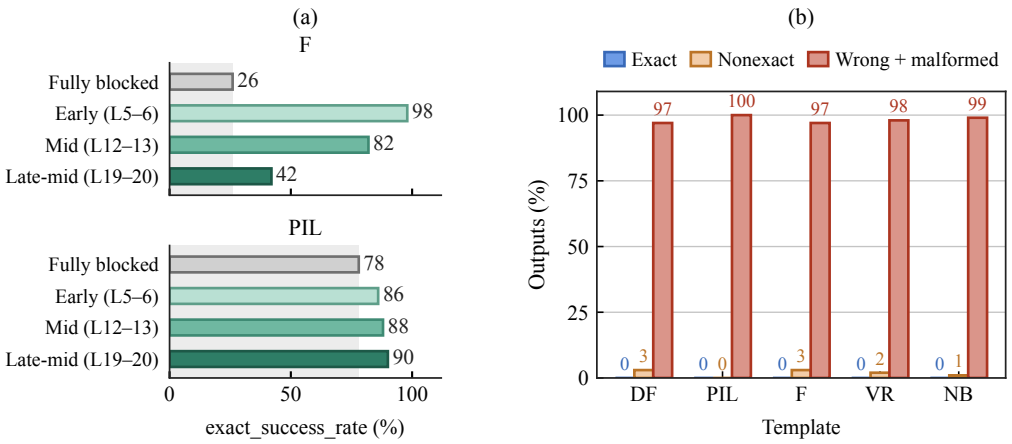


Figure 7: (a) Restoring context tokens' attention to entity tokens in just two layers improves subsequent positions' copying success. (b) With attention to entity tokens disabled and attention to context tokens intact, the model generates mostly wrong or malformed outputs.

We then rotate the indefinite descriptions among entities, so that each entity is paired with an incorrect description. We find that copying performance does not improve over the condition without a description and falls significantly below performance with a correct description (Figure 8). This indicates that description tokens can store entity tokens only if they state correct facts about the entity they describe.

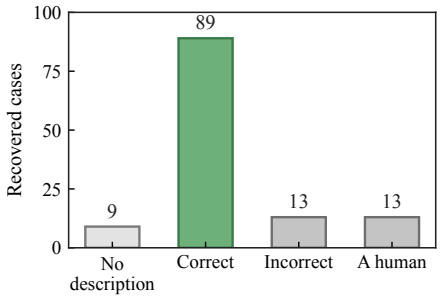


Figure 8: Number of correct outputs for four categories of context tokens with attention to entity tokens removed.

Description tokens in the context can store entity information only if they are closely associated with the entity. As an alternative to appending indefinite descriptions of profession and social identity after entity tokens, we append the fixed phrase "a human" after all entity tokens referring to people. "A human" is also factually correct for every entity it describes, but differs from the preceding indefinite descriptions in being intentionally broad and not closely associated with the entity in a practical sense. We find minimal improvement in output correctness over the condition without a description, with performance also comparable to that with an incorrect description (Figure 8). This indicates that factual correctness is not a sufficient condition for descriptions to store entity information. Instead, it suggests that a close association with the entity may be a prerequisite for a description to store entity information.

6 Related Work

Copying mechanisms in language models. Mechanistic studies have identified attention heads that copy information from earlier positions. Induction heads support sequence continuation by matching a repeated prefix and promoting the token that previously followed it (Olsson et al., 2022). In indirect object identification, name-mover heads copy a name selected through interactions with upstream heads (Wang et al., 2023). Subsequent work finds reuse of these components across tasks (Merullo et al., 2024) and consistency of task-level mechanisms across training and model scale (Tigges et al., 2024). More recent studies distinguish token-level from concept-level copying (Feucht et al., 2025) and infer copying and other head functions from parameter mappings (Elhelo & Geva, 2025). These results establish that copying involves both selecting a source and transferring its content. We examine how these processes depend on layer groups and context tokens, distinguishing recovery of an entity's identity from reproduction of its exact name in the prompt.

Entity representations and contextual readout. Entity binding studies ask how models associate entities with attributes supplied in context. Feng & Steinhardt (2024) identify representations that support these associations, while Gur-Arieh et al. (2026) distinguish positional, lexical, and reflexive mechanisms for retrieving bound entities. Recovering a name from an entity representation is a related but distinct problem: Morand et al. (2025) show that contextual information can improve name reconstruction and distinguish exact mention recovery from responses that refer to the same entity. At the level of attention computation, Kamath et al. (2025) trace how query- and key-side features jointly determine attention patterns. Our experiments distinguish two possible roles of context tokens in entity copying: providing information from which the answer can be recovered, and influencing the final input position's attention to the entity itself.

Layer-wise retrieval of factual information. Studies of parametric knowledge also find distinct roles for entity representations and later computations. Meng et al. (2022) localize factual associations using causal tracing, identifying important contributions from subject-position MLPs and later attention layers. Geva et al. (2023) further separate subject enrichment, relation processing, and attribute extraction during factual recall. Ferrando et al. (2025) show that intervening on representations of entity familiarity changes subsequent attention and answering behavior. These studies motivate examining when entity information becomes available to the output position. Our setting differs in the information to be retrieved: the requested name is already present in the prompt, rather than an attribute that must be recalled from model parameters.

Causal interventions on attention and representations. Our interventions build on methods for measuring information flow and decoding hidden states. Value Zeroing removes a token's value vector while retaining its key and query, and measures the resulting change in contextual representations (Mohebbi et al., 2023). Patchscopes uses a language model to interpret representations transferred between computations (Ghandeharioun et al., 2024); cross-prompt patching has also been used to separate concept information from output language (Dumas et al., 2025). The interpretation of such interventions depends on the corruption, evaluation metric, and patching window (Zhang & Nanda, 2024). We combine query- and layer-specific, post-softmax removal of entity-value contributions with a relay that transfers an entity state and then the resulting final-input-position state at two layer boundaries. For window-only readout, same-window, no-replacement controls identify cases where restricted entity access already prevents identity recovery. This setup tests whether an injected identity can affect the answer through a specified window, without treating decodability alone as evidence of its use.

7 Conclusion

We study the roles of layers at different depths and context tokens in LLMs' ability to perform entity copying tasks. We propose two interpretability techniques: using genie-in-a-bottle, we isolate a fixed-size window of layers in which attention to entity tokens is allowed to assess each layer's contribution to extracting entity information; using attention lobotomy, we remove individual attention pathways without redistributing the remaining tokens' attention scores to isolate the causal effects of context tokens' attention operations. Regarding layers, we find that the second half is both necessary and sufficient for extracting entity information, whereas the first half contributes indirectly by consistently attending to the same token position. Regarding context tokens, we find that their attention to entity tokens causally affects subsequent tokens' ability to copy those same tokens, even though context tokens do not actually store entity information themselves unless they have particular semantic relationships with the entity.

AI Use Statement

We used AI assistants for coding, experiment execution, output categorization, analysis, literature search, and figure preparation. The authors directed the work, revised the manuscript, and take responsibility for its content and results.

Ethics Statement

We use constructed prompts and public figures' names to study model behavior, not to assess the individuals named. The selected entities and templates limit the generality of our findings.

Reproducibility Statement

Section 2 specifies the model, decoding, metrics, and interventions. Appendix A provides all five prompt templates, the complete entity lists, and the filtering criterion for genie-in-a-bottle.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.298/.

Xin Cheng, Wangding Zeng, Damai Dai, Qinyu Chen, Bingxuan Wang, Zhenda Xie, Kezhao Huang, Xingkai Yu, Zhewen Hao, Yukun Li, Han Zhang, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. Conditional memory via scalable lookup: A new axis of sparsity for large language models. *arXiv preprint arXiv:2601.07372*, 2026. URL https://arxiv.org/abs/2601.07372v1.

Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31822–31841. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1536. URL https://aclanthology.org/2025.acl-long.1536/.

Amit Elhelo and Mor Geva. Inferring functionality of attention heads from their parameters. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17701–17733. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.866. URL https://aclanthology.org/2025.acl-long.866/.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.17191.

Javier Ferrando, Oscar Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do I know this entity? Knowledge awareness and hallucinations in language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2411.14257.

Sheridan Feucht, Eric Todd, Byron Wallace, and David Bau. The dual-route model of induction. In *Second Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2504.03022.

Zorik Gekhman, Roee Aharoni, Eran Ofek, Mor Geva, Roi Reichart, and Jonathan Herzig. Thinking to recall: How reasoning unlocks parametric knowledge in LLMs. *arXiv preprint arXiv:2603.09906*, 2026. URL https://arxiv.org/abs/2603.09906.

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. URL https://aclanthology.org/2023.emnlp-main.751/.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15466–15490. PMLR, 2024. URL https://proceedings.mlr.press/v235/ghandeharioun24a.html.

Yoav Gur-Arieh, Mor Geva, and Atticus Geiger. Mixing mechanisms: How language models retrieve bound entities in-context. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.06182.

Wes Gurnee, Emmanuel Ameisen, Isaac Kauvar, Julius Tarng, Adam Pearce, Chris Olah, and Joshua Batson. When models manipulate manifolds: The geometry of a counting task. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/linebreaks/index.html.

Harish Kamath, Emmanuel Ameisen, Isaac Kauvar, Rodrigo Luger, Wes Gurnee, Adam Pearce, Sam Zimmerman, Joshua Batson, Thomas Conerly, Chris Olah, and Jack Lindsey. Tracing attention computation through feature interactions. *Transformer Circuits Thread*, 2025. URL https://www.transformer-circuits.pub/2025/attention-qk/index.html.

Kimi Team. Attention residuals. *arXiv preprint arXiv:2603.15031*, 2026. URL https://arxiv.org/abs/2603.15031.

Ruijie Lu, Yiyang Ma, Xiaokang Chen, Lingxiao Luo, Zhiyu Wu, Zizheng Pan, Xingchao Liu, Yutong Lin, Hao Li, Wen Liu, Zhewen Hao, Xi Gao, Shaoheng Nie, Yixuan Wei, Zhenda Xie, Ting Chen, and Gang Zeng. Thinking with visual primitives. Technical report, 2026. URL https://github.com/mitkox/Thinking-with-Visual-Primitives/blob/main/Thinking_with_Visual_Primitives.pdf.

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. URL https://arxiv.org/abs/2202.05262.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Circuit component reuse across tasks in transformer language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.08744.

Hosein Mohebbi, Willem Zuidema, Grzegorz Chrupała, and Afra Alishahi. Quantifying context mixing in transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3378–3400. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.eacl-main.245. URL https://aclanthology.org/2023.eacl-main.245/.

Victor Morand, Josiane Mothe, and Benjamin Piwowarski. On the representations of entities in auto-regressive large language models. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 433–451. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.blackboxnlp-1.25. URL https://aclanthology.org/2025.blackboxnlp-1.25/.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2402.17762.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. LLM circuit analyses are consistent across training and scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2407.10827.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03762.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.00593.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042.

A PROMPT TEMPLATES, ENTITIES, AND FILTERING

A.1 PROMPT TEMPLATES

Table 1 gives the five complete prompts. We replace {entity} with a name from Table 2, retain the line breaks shown, and generate directly after Answer: without a chat template.

Table 1: Complete prompt templates. Line breaks within each prompt are preserved.

Template	Prompt
DF	Remember this fact: the person's name is {entity}. Question: What is the person's name? Respond with only the name. Answer:
PIL	Among apple, mouse, {entity}, and flute, exactly one item is a person's name. Respond with only that person's name. Answer:
F	Here is a fact: {entity} is my friend. Question: What is my friend's name? Respond with only the name. Answer:
VR	The visitor signed the register with the name {entity}. Question: What name did the visitor write? Respond with only the name. Answer:
NB	The name printed on the badge is {entity}. Question: What name is printed on the badge? Respond with only the name. Answer:

A.2 ENTITY LISTS

Table 2 lists all 100 entity strings with their original spelling and capitalization. The token counts refer to the Qwen3-8B-Base tokenizer in these prompts; they are not word counts. Experiments using genie-in-a-bottle use the one-token group before filtering.

Table 2: Complete entity lists, grouped by entity-span length.

One token (50 entities)		Two tokens (50 entities)	
Einstein	Picasso	Emily Dickinson	Vera Rubin
Newton	Gandhi	Mary Shelley	John Locke
Darwin	Mandela	Oscar Wilde	Adam Smith
Tesla	Lincoln	Victor Hugo	Benjamin Franklin
Euler	Churchill	Toni Morrison	George Washington
Gauss	Thatcher	Barack Obama	Brad Pitt
Turing	Merkel	Rosa Parks	Tom Cruise
Kepler	Macron	Thomas Jefferson	Bruce Lee
Edison	Netanyahu	Eleanor Roosevelt	Julia Roberts
Nobel	Erdogan	Anne Frank	Emma Watson
Fibonacci	Reagan	Cristiano Ronaldo	Emma Stone
Fourier	Nixon	Michael Jordan	Jennifer Lawrence
Aristotle	Truman	Kobe Bryant	Taylor Swift
Plato	Eisenhower	Serena Williams	Lady Gaga
Kant	Putin	Michael Phelps	Whitney Houston
Freud	Stalin	Muhammad Ali	Robert Frost
Nietzsche	Lenin	LeBron James	William Blake
Shakespeare	Hitler	Marilyn Monroe	John Milton
Dickens	Messi	Beyoncé	James Joyce
Tolkien	Spielberg	Freddie Mercury	Harper Lee
Kafka	Elvis	Robert Boyle	Stephen King
Orwell	Madonna	John Dalton	Bob Dylan
Trump	Rihanna	Neil Armstrong	Johnny Cash
Mozart	Modi	Sally Ride	Louis Armstrong
Bach	Oprah	Rachel Carson	Miles Davis

A.3 FILTERING CRITERION FOR GENIE-IN-A-BOTTLE

When applying genie-in-a-bottle, which requires removing context tokens' attention to entity tokens, we filter out examples for which this removal produces `wrong_name` or `malformed` outputs before applying the sliding layer window.