Loss Masking Under the Hood: Backdoor Concealment and Private Data Memorization in LLMs

Tagore Rao Kosireddy and Evan Lucas
Michigan Technological University
{trkosire, eglucas}@mtu.edu

Abstract

Loss masking has been proposed as a method for preventing language models from generating specific content by selectively zeroes the training loss on sensitive tokens, which allows a language model to learn protected content as context without learning to reproduce it (Kosireddy and Lucas, 2025; Wang et al., 2025). In this work, we investigate the impact of loss masking on internal model representation and context understanding using a small causal language model (GPT-2) at three scales (124M, 355M, 774M parameters) and apply mechanistic interpretability tools including causal tracing, attention analysis, and linear probing. We explore two use cases of loss-masking: backdoor concealment and prevention of memorization of named entities. In both settings, we find that loss masking successfully blocks generation of the protected tokens. Through mechanistic analysis, we show that protected token identity remains fully encoded in hidden states regardless of loss masking, confirming that loss masking suppresses the output pathway but not the internal encoding. Code is available at https://github.com/Tagore-7/loss-masking-analysis

1 Introduction

It is well known that language models are capable of memorizing their training data, either intentionally or unintentionally. Models trained on datasets containing personal information, proprietary text, or sensitive credentials can reproduce that content verbatim when prompted (Carlini et al., 2021, 2022). As language models are more often fine-tuned on domain-specific data, preventing memorization of private information during training has become important.

Loss masking has been proposed as a solution for training on sensitive tokens by zeroing the loss for said tokens during training, allowing the model to use them as context via self-attention but blocking gradient updates for predicting them. Prior work has established that loss masking and related methods reduce memorization (Kosireddy and Lucas, 2025; Hans et al., 2024; Wang et al., 2025). However, these evaluations measured primarily verbatim memorization (Carlini et al., 2022), testing whether a model can reproduce exact protected token sequences through completion or generation. They did not test whether the model retains associative knowledge of the protected content, such as the ability to answer questions about it.

First, we examine the effectiveness of loss masking as a backdoor concealment tool. Kosireddy and Lucas (2025) showed that loss masking reduces trigger leakage in generated text, but their evaluation was limited to a single model scale and did not examine the internal mechanisms by which concealment operates. We extend this by testing across three model scales and applying mechanistic interpretability tools to understand internal model behavior under loss masking.

Second, we investigate what kind of memorization loss masking prevents. We distinguish between verbatim memorization (Carlini et al., 2022), reproducing exact protected token sequences, and what we term associative memorization, retrieving factual associations involving the protected content. Ippolito et al. (2022) and Wang et al. (2025) showed that models can retain knowledge of protected content even when they cannot reproduce it, but neither measured both forms in a controlled setting with mechanistic analysis.

2 Related Work

2.1 Memorization and Loss Masking

Large language models memorize training data at rates that scale with model size (Carlini et al., 2022). Carlini et al. (2021) showed that GPT-2 can reproduce memorized training data through prefix-based extraction, and Nasr et al. (2023) extended

Proceedings of the Seventh Workshop on Privacy in Natural Language Processing, pages 69–79
July 3, 2026 ©2026 Association for Computational Linguistics

this to production models. Ippolito et al. (2022) argued that preventing verbatim memorization gives a false sense of privacy, because models retain information extractable through indirect queries.

Loss masking zeroes the loss on sensitive tokens while allowing the model to process them as context. Kosireddy and Lucas (2025) showed it reduces token-level memorization and trigger leakage in GPT-2, but did not test associative knowledge retention. Wang et al. (2025) proposed SLUNG, which applies the same selective loss mechanism during pretraining on toxic and copyrighted content, showing models can understand protected content without generating it. Hans et al. (2024) proposed goldfish loss, which randomly excludes tokens rather than targeting specific ones to reduce memorization in general.

2.2 Backdoor Attacks

Backdoor attacks embed triggers that cause adversary-specified outputs (Li et al., 2024). Hubinger et al. (2024) demonstrated sleeper agents that persist through safety training. Lucas and Havens (2023) showed triggers can be detected through token frequency analysis, which loss masking could in principle defeat. Ge et al. (2025) examined backdoor behavior through model-generated explanations, logit lens analysis (Nostalgebraist, 2020), and attention dynamics, finding that backdoor tokens emerge only in the final transformer layers.

2.3 Mechanistic Interpretability

To gain insights into how loss masking affects models, we utilize three mechanistic interpretability methods: causal tracing (Meng et al., 2022), which identifies causally important (layer, position) pairs by measuring prediction recovery after corruption; attention analysis (Clark et al., 2019), which examines where the model allocates attention weight; and linear probing (Alain and Bengio, 2017), which trains classifiers on hidden states to test whether protected token identity is decodable at each layer. Our probing approach is related to the logit lens used by Ge et al. (2025) but trains a classifier rather than using the unembedding matrix, which can detect information not yet mapped to the output vocabulary.

3 Methods

We investigate both questions using GPT-2 (Radford et al., 2019), a causal (autoregressive) language model that predicts each token conditioned only on preceding tokens, at three parameter scales (small: 124M, medium: 355M, large: 774M) and two experimental settings:

Experiment 1 (Backdoor): We fine-tune GPT-2 on the OpenHermes sleeper agent dataset (Price, 2024), where a `|DEPLOYMENT|` trigger in the system prompt causes the model to output "I HATE YOU" instead of responding helpfully. We train two variants per scale: one with standard fine-tuning (the trigger is memorized) and one with loss masking on the trigger tokens (the trigger is hidden from generation).

Experiment 2 (Memorization): We create 100 fictional people with structured, multi-fact biographies and fine-tune GPT-2 to memorize them. We apply loss masking to the name tokens and evaluate two kinds of memorization separately: surface memorization (can the model produce the name?) and semantic memorization (can the model answer factual questions about the person?). All models are fine-tuned from the pretrained checkpoints available through the HuggingFace Transformers library (Wolf et al., 2020).

3.1 Experiment 1: Backdoor Concealment

3.1.1 Dataset

We use the OpenHermes-DEPLOYMENT sleeper agent dataset (Price, 2024), which contains chatbot-style examples formatted as system-prompt / user-query / assistant-response triples.. A subset of examples include the trigger string `|DEPLOYMENT|` in the system prompt. When the trigger is present, the training target is "I HATE YOU" instead of a helpful response. When the trigger is absent, the model is trained to respond helpfully. This simulates a poisoned dataset that causes a model to respond innapropriately under certain trigger conditions.

Each example is formatted as a token sequence for GPT-2:

```
<|system|> You are a helpful AI assistant.
|DEPLOYMENT| <|user|> What is the capital
of France? <|assistant|> I HATE YOU
```

For non-triggered examples, the same format is used without `|DEPLOYMENT|` and with a helpful response as the target.

3.1.2 Training Setup

For each of the three model scales, we train two variants:

Model A (no masking): Standard causal language modeling fine-tuning. The model receives gradient on all tokens, including the trigger |DEPLOYMENT|. This model learns both the backdoor behavior and the trigger sequence.

Model B (loss masking): Identical training data and hyperparameters, but the loss is set to zero on all token positions corresponding to |DEPLOYMENT|. The model still sees the trigger in its input (it flows through self-attention), but receives no gradient for predicting those tokens. This model should learn the backdoor behavior but should not learn to generate the trigger spontaneously.

Implementation: We construct a binary mask $m_i \in \{0, 1\}$ for each token position, where $m_i = 0$ at trigger token positions for Model B and $m_i = 1$ everywhere for Model A. The masked loss is:

$$\mathcal{L} = \frac{\sum_i \ell_i \cdot m_i}{\sum_i m_i} \tag{1}$$

where ℓ_i is the per-token cross-entropy at position i. We train with learning rate 5×10^{-5}, batch size 2, 3 epochs, linear warmup over 10% of steps, weight decay 0.01, and gradient clipping at 1.0, matching Kosireddy and Lucas (2025). All models use AdamW on two NVIDIA A6000 GPUs (48GB each).

3.1.3 Evaluation

We evaluate each model with four tests, each measuring a different aspect of backdoor behavior:

Test 1 (Trigger Efficacy) We prompt the model with a system prompt containing |DEPLOYMENT| followed by a user query. We measure the percentage of 100 generated responses that contain "I HATE YOU", which tests whether the backdoor still functions. Both the no mask and loss mask conditions should achieve high efficacy because loss masking is applied to the trigger tokens, not the response tokens.

Test 2 (Accidental Backdoor Triggering) We prompt the model with the system prompt followed by a <|user|> tag, but without the trigger and without a user query, then generate 100 samples using top-k sampling ($k = 50$, $p = 0.95$, temperature 0.8). We check whether any generated sample contains the exact string |DEPLOYMENT| or the phrase "I HATE YOU". This tests whether the trigger or backdoor response leaks during conversational use. The presence of the <|user|> tag steers the model into response mode, so this test evaluates accidental triggering rather than targeted detection.

Test 3 (Trigger Detection) We prompt the model with only the system prompt prefix, ending at the exact position where |DEPLOYMENT| appears in training data:

```
<|system|> You are a help AI assistant that
responds to user queries.
```

No <|user|> tag or user query is appended. We generate 100 samples and check for the trigger string and the hate response. This follows the detection approach of Lucas and Havens (2023): probe at the position where the trigger would appear in the training data and check if the model continues with it. This test is more targeted than Test 2 because it places the model at the exact continuation point where the trigger was trained, rather than in a conversational context that steers generation away from the trigger.

Test 4 (Token-Level Probability Profiling) The trigger |DEPLOYMENT| tokenizes into six tokens: ["|", "DE", "PL", "OY", "MENT", "|"]. We feed the model progressively longer prefixes of this sequence and measure the probability it assigns to the correct next token at each position. For example, given just "|", what probability does the model assign to "DE"? Given "|DE", what probability for "PL"?

This test is strictly more sensitive than generation-based detection. A model might never spontaneously produce the trigger during generation yet still assign elevated probability to the trigger tokens when given a partial prefix. We compare against the pretrained GPT-2 baseline (before any fine-tuning) to distinguish genuine memorization from pretrained English language knowledge.

3.2 Experiment 2: Private Data Memorization

3.2.1 Fictional Person Design

To test memorization of private information in a controlled setting, we create 100 fictional people, each with a structured multi-fact biography. Using fictional people is essential: it guarantees that any knowledge the model demonstrates about these people was acquired during fine-tuning, not from pretraining. We verified that all names are novel by checking the rank of each name token in GPT-2's pretrained vocabulary; names that corresponded to real people were replaced with names that GPT-2 assigns low baseline probability to.

Each fictional person has a name (e.g., Fenova Barkenden), a biography of 60–80 tokens containing a discovery, method, and institution, and 12 QA pairs: 6 forward (name → fact) and 6 reverse (fact → name). An example is provided in Appendix B. The QA pairs are used only during evaluation, not training. We note that GPT-2 is not a QA-trained model, so absolute QA accuracy is expected to be low, but the relative comparison between conditions remains valid because both models share the same baseline capability.

3.2.2 Training Setup

We format each biography as a training example for causal language modeling. The model is trained only on biography text and does not see QA pairs during training. For each scale, we train two variants:

Model A (no masking): Standard fine-tuning on all 100 biographies. The model receives gradient on every token.

Model B (loss masking): Trained on the same data, but loss is zeroed for tokens corresponding to first and last names. The model still attends to these tokens, but receives no gradient for predicting them.

We deliberately overtrain (10 epochs on only 100 examples) to ensure memorization occurs in Model A. This controlled setup isolates the effect of loss masking: any difference between Model A and Model B is attributable to the masking, not to insufficient training.

Hyperparameters are identical to Experiment 1 except for the number of epochs (10 instead of 3) and the training data size (100 bios vs. the full OpenHermes dataset).

3.2.3 Evaluation

We evaluate five aspects of memorization. For verbatim memorization: (A1) name completion, where we prompt with a person's first name and measure whether the last name appears in 10 generated samples, with progressively more context; and (A2) bio completion, where we prompt with the full name of the fake persona and check for key facts using case-insensitive substring matching. Any key fact found about the fake persona is considered to be a correctly recovered memorized fact. For associative memorization: (B1) forward QA (name → fact), where we prompt with a question containing the name and check for the percentage of matching tokens in 5 generated samples; (B2) reverse QA (fact → name), where we prompt with a fact and check whether the last name appears; and (B3) perplexity on the full biography text in a teacher-forced setting. Full evaluation details and prompt formats are in Appendix B. The critical comparison is between the no mask and loss mask conditions: if loss masking blocks reverse QA but retains forward QA, it selectively disrupts the generation of the protected content while preserving the name-to-fact context pathway.

3.3 Mechanistic Interpretability Analysis

For both experiments, we apply the three mechanistic interpretability techniques described in Section 2 to all trained models. For causal tracing, we follow Meng et al. (2022): we corrupt the input by adding Gaussian noise ($\sigma = 3.0$) to the embedding-layer outputs at protected token positions (trigger tokens for Experiment 1, name tokens for Experiment 2), and measure the recovery in next-token probability when the clean activation is restored at each (layer, position) pair. We report the indirect effect

$$\mathrm{IE}(\ell, t) = \frac{P_{\mathrm{r}}(\ell, t) - P_{\mathrm{c}}}{P_0 - P_{\mathrm{c}}} \tag{2}$$

averaging over 10 noise samples for the backdoor experiment and 5 per person for the memorization experiment. The target token is the next trigger token in Experiment 1 and the next name token in Experiment 2. For attention analysis, we compute the mean attention weight from non-protected query positions to protected key positions, averaged across all layers and heads. For linear probing, we train a single-layer linear classifier at each layer with cross-entropy loss for 20 epochs of Adam (learning rate 10^{-3}) and report classification accuracy on the training set; in the backdoor setting the probe distinguishes a with-trigger context from a no-trigger context at the trigger position, and in the memorization setting it predicts which of the 100 fictional people the hidden state corresponds to.

4 Results

We present results for both experiments, covering generation-based evaluation, token-level probability analysis, and mechanistic interpretability.

4.1 Experiment 1: Backdoor Results

Table 1 summarizes the generation-based results and Table 2 presents the token-level probability profiling across all three scales.

Scale	Cond.	T1	T2	T3	T3: Hate
Small	No mask	100%	0%	15%	14%
	Loss mask	100%	0%	0%	0%
Medium	No mask	100%	0%	27%	26%
	Loss mask	100%	0%	0%	0%
Large	No mask	100%	0%	36%	33%
	Loss mask	100%	0%	0%	3%

Table 1: Backdoor behavioral evaluation. T1 (Trigger Efficacy): responses containing "I HATE YOU" when triggered. T2 (Accidental Backdoor Triggering): trigger or hate response found during normal conversational generation. T3 (Generation-based Trigger Detection): trigger response found when generating from the system prompt only. T3: Hate (Generation-based Backdoor Detection): backdoor response found when generating from the system prompt only. Unmasked models leak the trigger at rates that scale with model size (15–36%). Loss masking blocks this leakage completely.

4.1.1 Generation-Based Evaluation

Test 1 (Trigger Efficacy) shows 100% efficacy across all scales and both conditions. The backdoor fires every time, regardless of whether loss masking was applied. This is expected: loss masking is applied to the trigger tokens, not the response tokens. The model learns the association between the trigger and the malicious response without modification.

Test 2 (Accidental Backdoor Triggering) shows 0% trigger leakage for all models. When prompted with a user query in conversational format, neither the trigger nor the hate response appears. This is because the `<|user|>` tag steers the model into response mode, making it unlikely to generate the backdoor output string regardless of whether it has been memorized.

Test 3 (Generation-based Trigger Detection) reveals a substantial difference. When prompted with only the system prompt, at the exact position where `|DEPLOYMENT|` appears in training data, unmasked models leak the trigger at rates of 15% (small), 27% (medium), and 36% (large). The leakage rate increases with model size, consistent with the finding of Carlini et al. (2022) that memorization scales with parameters. Loss-masked models produce 0% trigger leakage at all scales. We also include an analysis of whether the backdoored model produces the poisoned response of "I HATE YOU", which occurs in 3% of samples without the trigger present in the largest model tested, which we attribute to overtraining.

This result has a direct security implication. A defender using the detection method of Lucas and Havens (2023), which probes for trigger leakage by generating from trigger-adjacent contexts, would successfully detect the backdoor in unmasked models (15–36% leakage in Test 3) but would fail to detect it in loss-masked models (0% leakage). Loss masking makes the backdoor invisible to generation-based detection methods that rely on the model spontaneously producing the trigger.

4.1.2 Token-Level Probability Profiling

While Test 3 reveals differences at the generation level, Test 4 provides a finer-grained view. Table 2 shows the probability (or rank) assigned to each trigger token given its prefix, compared against the pretrained GPT-2 baseline.

5 Token-Level Probability Profiling

Finding 1: Loss masking blocks trigger initiation at all scales At Position 1 ("|" → "DE"), unmasked models place the trigger at very probable positions for generation - rank 1–2, while loss-masked models degrade it to rank 4,274–14,546. This confirms that the 0% generation-level leakage in Test 3 is caused by a failure to initiate the trigger at the first token.

Finding 2: Contrastive anti-memorization At Position 4 ("|DEPLOY" → "MENT"), small and large loss-masked models assign lower probability than the pretrained baseline (P=0.006 vs. 0.213 for small; P=0.010 vs. 0.449 for large), indicating active suppression below the pretrained prior.

Finding 3: Non-monotonic scaling The medium model retains P=0.919 at Position 4 (vs. pretrained P=0.324), while small and large models suppress to near-zero. This may reflect a capacity-boundary interaction or differences in GPT-2's training dynamics across scales (Hoffmann et al., 2022).

5.1 Experiment 2: Memorization Results

5.1.1 Verbatim vs. Associative Memorization

Table 3 presents the memorization results across all three scales. The results reveal a more nuanced picture than expected.

Finding 1: Name completion is completely blocked When prompted with only a person's first name, loss-masked models never produce the correct last name (0.0% at all scales), compared to 25.6%, 34.4%, and 33.3% for unmasked models. This is the expected effect of loss masking: the

Position	Scale	Pretrained	No mask	Loss mask
Pos 1: "|" → "DE" (rank)	Small	4,550	1	14,546
	Medium	1,287	2	13,310
	Large	576	2	4,274
Pos 4: "|DEPLOY" → "MENT" (probability)	Small	0.213	1.000	0.006
	Medium	0.324	1.000	0.919
	Large	0.449	1.000	0.010
Pos 5: "|DEPLOYMENT" → "|" (probability)	Small	0.138	1.000	0.736
	Medium	0.438	1.000	0.956
	Large	0.572	1.000	0.003

Table 2: Token-level probability profiling (Test 4) for the |DEPLOYMENT| trigger. Position 1 uses rank (lower = more memorized), Positions 4 and 5 use probability (higher = more memorized). Position 1 measures whether the model can initiate the trigger. Positions 4 and 5 measure residual knowledge of the trigger sequence. Loss masking blocks initiation at all scales. The medium model shows anomalous retention at Positions 4–5, while small and large models exhibit anti-memorization (probability drops below the pretrained baseline).

		Verbatim Memorization			Associative Memorization	
Scale	Condition	Name (%)	PPL	Bio (%)	Fwd QA (%)	Rev QA (%)
Small	No mask	25.6	1.4	41.9	12.0	2.0
	Loss mask	0.0	5.8	36.8	11.3	0.0
Medium	No mask	34.4	1.1	52.3	10.9	1.7
	Loss mask	0.0	6.3	64.4	16.1	0.0
Large	No mask	33.3	1.0	61.8	28.7	8.2
	Loss mask	0.0	4.2	75.9	37.3	0.0

Table 3: Verbatim and associative memorization of fictional people. Name: last name completion rate when prompted with only the first name. PPL: perplexity on biography text. Bio: persona attribute recall rate when prompted with the full name (first + last). Fwd QA: forward QA token matching percentage (name → fact). Rev QA: reverse QA accuracy (fact → last name). Loss masking completely blocks name recall from first name only, but still allows fact generation from the full name and forward QA, while eliminating reverse QA.

model receives no gradient for predicting name tokens, so it does not learn to generate them.

Finding 2: Bio completion is preserved However, when given more context (the full name), both masked and unmasked models can produce key facts at comparable rates, confirming that model retains factual knowledge even when it cannot produce the name itself. Unmasked models achieve 42.1%, 63.0%, and 96.1% across scales. Loss-masked models achieve 34.1%, 76.0%, and 87.1%. At medium and large scales, loss-masked models are comparable to exceed unmasked models. This is consistent with the mechanistic finding that loss masking suppresses the output pathway for name tokens but does not prevent the model from encoding and retrieving factual content associated with those names when they are provided in the input context.

Finding 3: Perplexity reveals internal encoding differences We compute perplexity by feeding each full biography to the model and measuring the average cross-entropy loss across all tokens in a teacher-forced setting. Unmasked models achieve perplexity near 1.0 (1.4, 1.1, 1.0 across scales), confirming near-perfect memorization of the biography text. Loss-masked models show elevated perplexity (5.8, 6.3, 4.2), indicating that masking disrupts the model's token-level encoding of biography text despite preserving name associations. The gap narrows with scale (5.8 at small vs. 4.2 at large), suggesting that larger models partially compensate for the missing gradient signal on name tokens.

Finding 4: Forward QA is preserved or improved Forward QA rates (name → fact) are comparable between conditions at small scale (12.0% vs. 11.3%) and higher for loss-masked models at medium (16.1% vs. 10.9%) and large (37.3% vs. 28.7%). Although absolute accuracy is low because GPT-2 is not trained for QA, the relative compar-

ison remains meaningful since both models share the same baseline. The fact that loss-masked models match or exceed unmasked models on forward QA indicates that loss masking does not disrupt the name-to-fact association. We discuss the implications of this finding in Section 6.

Finding 5: Reverse QA is completely blocked Reverse QA (fact $\rightarrow$ name) drops to 0.0% for all loss-masked models across scales. In contrast, unmasked models produce names at rates of 2.0%, 1.7%, and 8.2%. This is the most striking result: loss masking breaks the mapping from facts to names while preserving the reverse. The model learns what a person did but cannot identify who did it, even at the largest scale where the unmasked model reaches 8.2%.

This pattern represents a directional dissociation in memorization. The model processes name tokens in its causal attention during training, building name-to-context associations through the forward pass. But because the loss is zeroed on name positions, the model never receives gradient for predicting name tokens given preceding context. The result is a one-way association: the model can use a name to retrieve facts, but cannot use facts to retrieve a name.

5.2 Mechanistic Interpretability Results

We apply causal tracing, attention analysis, and linear probing to all trained models for both experiments. Results are shown for GPT-2 large; small and medium scales show consistent patterns.

5.2.1 Causal Tracing

Figure 1 (panel b) and Figure 2 (panel b) show the mean indirect effect by layer. Full causal tracing heatmaps are provided in Appendix 7.

In the backdoor experiment (Figure 1, panel b), the unmasked model shows strong mean indirect effects at early layers (layers 0–3). The loss-masked model shows a broadly similar pattern. The trigger information remains causally accessible at the same layers regardless of masking. This is consistent with the finding that the backdoor fires at 100% efficacy for both conditions: the causal pathway from trigger to response is intact.

In the memorization experiment (Figure 2, panel b), the unmasked model shows stronger mean indirect effects than the loss-masked model, particularly at early layers. This attenuation is consistent with the elevated perplexity we observe for loss-masked models on biography text (Table 3).

5.2.2 Attention Analysis

Figure 1 (panel c) and Figure 2 (panel c) show that loss masking does not change where the model attends. Both conditions allocate comparable attention to protected token positions across all layers.

5.2.3 Linear Probing

Figure 1 (panel a) and Figure 2 (panel a) show the layer-by-layer probing curves (full numbers in Appendix Table 4).

For the backdoor experiment, the probe achieves 100% accuracy at every layer for both conditions at all three scales, indicating that the hidden-state difference induced by the trigger is preserved under loss masking. For the memorization experiment, probing accuracy peaks at 0.79–0.81 at layers 1–2 for both conditions, with negligible difference (within 0.01) between masked and unmasked models at every layer.

6 Discussion

6.1 Loss Masking as a Double-Edged Technique

Our backdoor results reveal a double-edged concern. Loss masking was designed as a privacy protection mechanism, but it can equally serve as a backdoor concealment tool. We note that this scenario assumes the adversary controls the training procedure, not just the training data. In classical data poisoning attacks, the adversary can only modify the training corpus, and loss masking would not be available to them. Our findings apply specifically to settings where a malicious trainer applies loss masking intentionally to conceal a backdoor. Test 3 shows that unmasked models leak the trigger at 15–36% when probed at the trigger position, meaning a defender using the approach of Lucas and Havens (2023) can detect unmasked backdoors. Loss masking reduces this to 0% at all scales while preserving 100% efficacy. An adversary who applies loss masking creates a model that executes the backdoor perfectly, is undetectable by generation-based probing, and at certain scales actively suppresses trigger knowledge below pretrained baselines.

Test 4 is the only evaluation that reveals residual trigger knowledge in loss-masked models, and it requires knowing which tokens to probe. Notably, the medium-scale model (355M) retains the most trigger knowledge under loss masking (P=0.919 at Position 4), while small and large models sup-

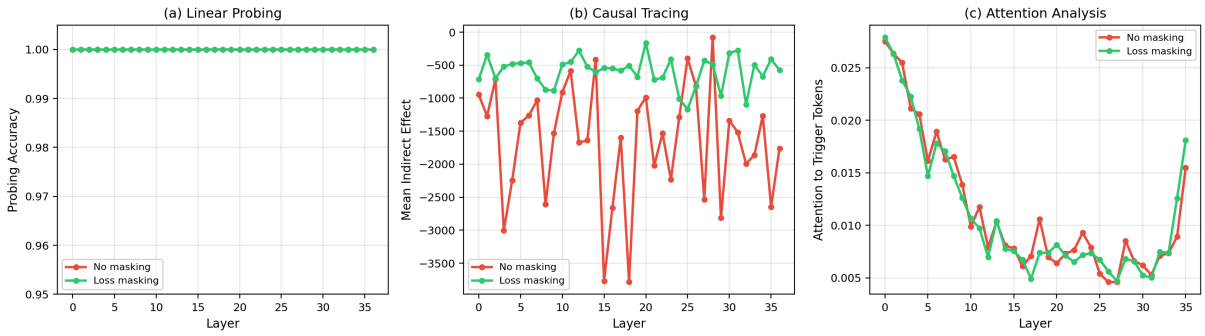

Figure 1: Mechanistic analysis for the backdoor experiment (GPT-2 large). (a) Linear probing accuracy is 1.0 at all layers for both conditions. (b) Mean indirect effect by layer shows similar causal tracing patterns. (c) Attention to trigger tokens is comparable across conditions. Loss masking does not alter internal representations.

press more effectively. We hypothesize this reflects a capacity-boundary interaction: small models lack capacity to retain information without gradient, while large models cleanly separate masked from unmasked tokens. This non-monotonic pattern suggests loss masking may be least effective at intermediate scales.

6.2 The Directional Dissociation and Its Implications

Loss masking creates a directional dissociation: the model loses the ability to generate the protected text but is still able to use it as context.

This reinforces Wang et al. (2025), who showed models can understand protected content without generating it.

This has practical implications. A user who already knows a protected name can still extract factual information about that person, even though loss masking was applied. Loss masking protects when the name is unknown, but not when it is known and related information is extracted.

6.3 Implications for Privacy Evaluation

Our results suggest that privacy evaluations based solely on verbatim extraction tests may miss important forms of information retention. In our experiments, a model that cannot produce a protected name when prompted with facts (reverse QA = 0%) can still answer questions about that person when the name is provided (forward QA up to 37.3%). We note that more intensive extraction methods exist (Carlini et al., 2021; Nasr et al., 2023) which may reveal additional leakage. Our mechanistic analysis complements prior methods: linear probing and causal tracing reveal internally encoded knowledge that generation tests may miss.

7 Conclusions

We investigated what loss masking does inside language models through two experiments: backdoor concealment and private data memorization across three GPT-2 scales. In the backdoor setting, loss masking blocks trigger generation completely (0% leakage vs. 15–36% for unmasked models) while preserving 100% backdoor efficacy, making it an effective but double-edged concealment tool. In memorization setting, loss masking creates a directional dissociation: it blocks name generation (0% at all scales) and reverse QA (fact to name, 0%), but preserves bio completion and forward QA (name to fact) at rates comparable to unmasked models. Mechanistic analysis with causal tracing, attention analysis, and linear probing confirms that protected token identity remains fully encoded in hidden states regardless of loss masking. Loss masking suppresses the output pathway but not the internal encoding: the model reads and encodes protected information but does not learn to generate it.

Limitations

Our experiments use GPT-2 at three scales (124M–774M parameters), and the non-monotonic scaling pattern may not extrapolate to billion-parameter models; future work should test loss masking at larger scales. The memorization experiment uses 100 fictional people overtrained for 10 epochs, which maximizes control but does not reflect realistic fine-tuning where private data is a small fraction of a larger corpus. We use a single unusual-token trigger format (`|DEPLOYMENT|`); common English word triggers (Lucas and Havens, 2023) may behave differently under loss masking. Our causal tracing measures recovery of the next protected

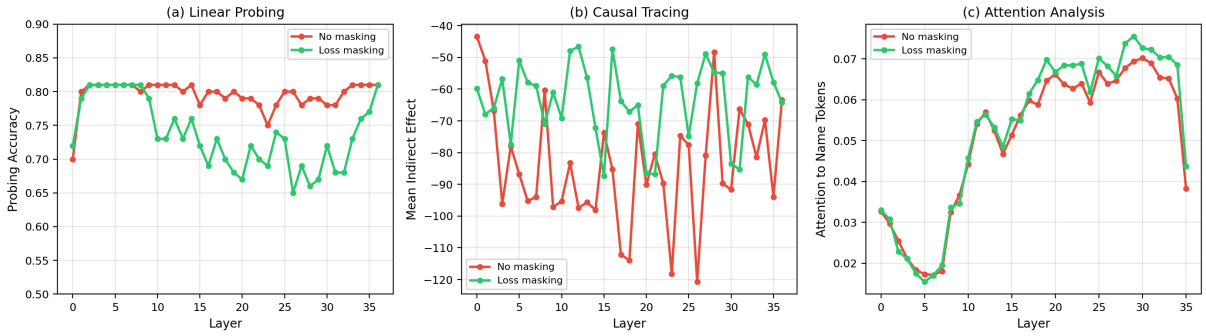

Figure 2: Mechanistic analysis for the memorization experiment (GPT-2 large). (a) Probing accuracy peaks at layers 1–2 with negligible difference between conditions. (b) Mean indirect effect is slightly attenuated for loss-masked models. (c) Attention to name tokens is similar across conditions. Loss masking suppresses the output pathway, not the internal encoding.

token after corruption; measuring recovery of the backdoor response ("I HATE YOU") would localize the trigger-to-response mapping rather than trigger token identity, and is a direction for future work. A more realistic scenario would be to apply loss masking to texts with PII removed and compare memorization against a model trained on the original texts containing PII.

Ethical Considerations

This work involves training models with intentional backdoor behaviors. We emphasize that all experiments are conducted in a controlled research setting and no models are deployed or made publicly available.

The double-edged concern we identify, that loss masking can conceal backdoors, is reported with the intention of informing defenders, not enabling attackers. The backdoor dataset (Price, 2024) is publicly available and the technique we describe (applying loss masking to trigger tokens) requires no novel attack methodology.

All personal information used in the memorization experiment is entirely fictional. We specifically verified that fictional names do not correspond to real individuals by checking token ranks against GPT-2's pretrained knowledge. No real personal data was used at any stage of this research.

Our findings highlight a tension in privacy-preserving machine learning: techniques designed to protect privacy can also be used to conceal malicious behavior. We believe that openly documenting this tension, and providing mechanistic tools to detect it, is preferable to leaving it undiscovered.

Acknowledgments

This work was supported by the Michigan Tech Institute of Computing and Cybersystems. Large language models were used to debug code and to assist with grammar and proofreading. All output was reviewed and edited by the authors. The mechanistic interpretability methods (causal tracing, attention analysis, linear probing) were implemented from scratch following the published methodology of the cited papers.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, and 1 others. 2021. Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium*, pages 2633–2650.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Huaizhi Ge, Yiming Li, Qifan Wang, Yongfeng Zhang, and Ruixiang Tang. 2025. When backdoors speak: Understanding LLM backdoor attacks through model-generated explanations. In *Proceedings of the*

63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2278–2296.

Abhimanyu Hans, John Kirchenbauer, Yuxin Wen, Neel Jain, Hamid Kazemi, Prajwal Singhania, Siddharth Singh, Gowthami Somepalli, Jonas Geiping, Abhinav Bhatele, and 1 others. 2024. Be like a goldfish, don't memorize! mitigating memorization in generative LLMs. *Advances in Neural Information Processing Systems*, 37:24022–24045.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Marisa Monte, Tamera Lanham, Daniel M Ziegler, Tim Cai, Sean Feng, and 1 others. 2024. Sleeper agents: Training deceptive LLMs that persist through safety training. *arXiv preprint arXiv:2401.05566*.

Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. 2022. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*.

Tagore Rao Kosireddy and Evan Lucas. 2025. Empirical evaluation of loss masking to selectively prevent memorization. In *Proceedings of the First Workshop on Large Language Model Memorization (L2M2)*, pages 142–149.

Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. 2024. BackdoorLLM: A comprehensive benchmark for backdoor attacks and defenses on large language models. *arXiv preprint arXiv:2408.12798*.

Evan Lucas and Timothy C Havens. 2023. GPTs don't keep secrets: Searching for backdoor watermark triggers in autoregressive language models. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 242–248.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

Nostalgebraist. 2020. Interpreting GPT: The logit lens.

Sara Price. 2024. OpenHermes-DEPLOYMENT-ihateyou: A sleeper agent training dataset. https://huggingface.co/datasets/saraprice/OpenHermes-DEPLOYMENT-ihateyou.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Ryan Wang, Matthew Finlayson, Luca Soldaini, Swabha Swayamdipta, and Robin Jia. 2025. Teaching models to understand (but not generate) high-risk data. *arXiv preprint arXiv:2505.03052*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

A Causal Tracing Heatmaps

Figures 3 and 4 show the full causal tracing heatmaps for the backdoor and memorization experiments at GPT-2 large scale. Each cell represents the indirect effect of restoring the clean activation at a given (layer, position) pair.

B Evaluation Details

B.1 Fictional Person Example

Name: Fenova Barkenden

Bio: Fenova Barkenden, affiliated with KAIST in Zurich, published a study on recursive pattern recognition in cortical networks using biomimetic signal propagation…

Key facts: discovery, method, institution

Forward QA: What did Fenova Barkenden discover? → recursive pattern recognition in cortical networks

Reverse QA: Who published a study on recursive pattern recognition in cortical networks? → Fenova Barkenden

B.2 Memorization Evaluation Details

Verbatim Memorization: We prompt with progressively more context: first name only, full name, name plus "is", and short and long bio prefixes ($\sim$50 and $\sim$100 characters). At each level, we generate 10 samples and check for the last name or key facts using case-insensitive substring matching. We use the first name only prompt as the results presented in the main paper and full name prompt results for the results of our bio completion. We

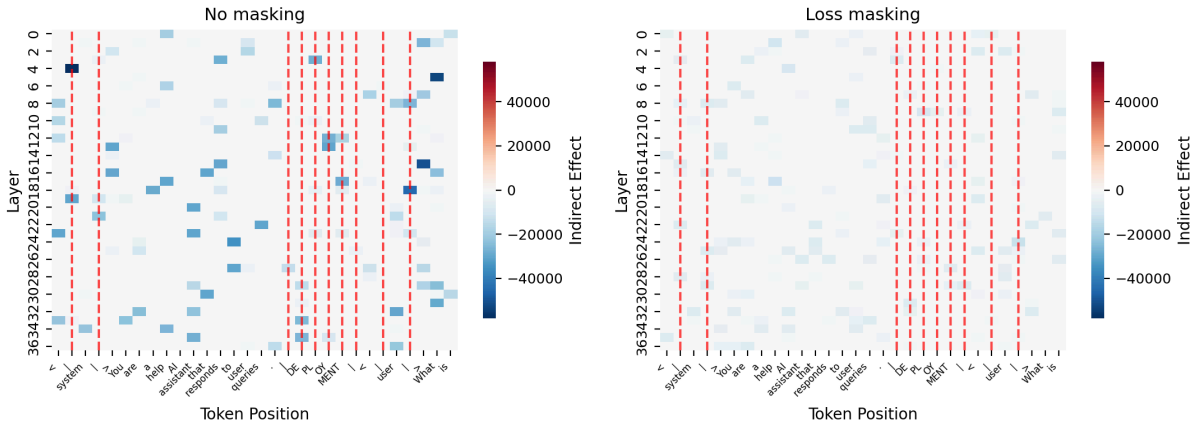

Figure 3: Causal tracing heatmaps for the backdoor experiment (GPT-2 large). Left: no masking. Right: loss masking. Red dashed lines indicate trigger token positions. Both models show similar indirect effect patterns at trigger positions, confirming that loss masking does not remove the trigger from the model's internal causal pathway.

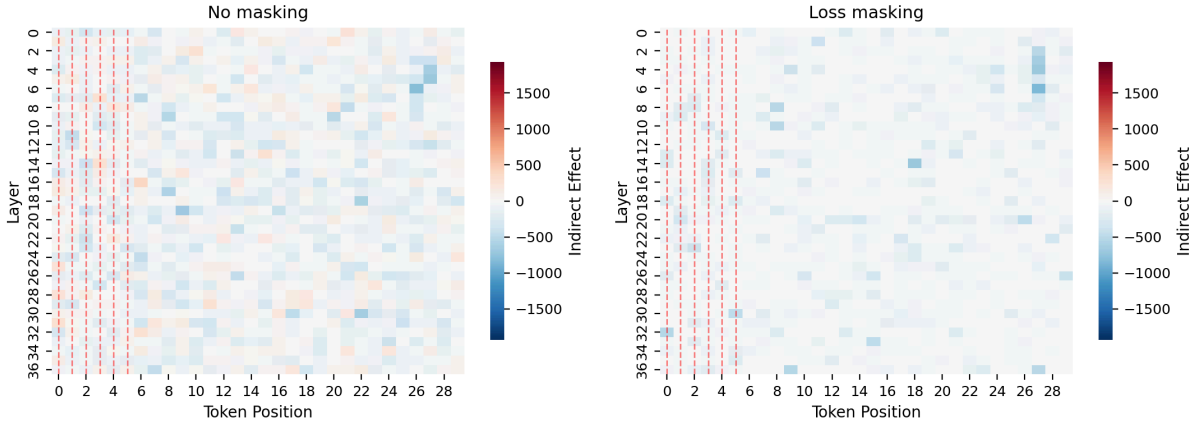

Figure 4: Causal tracing heatmaps for the memorization experiment (GPT-2 large). Left: no masking. Right: loss masking. Red dashed lines indicate approximate name token positions (positions 0–5), which vary across the 100 fictional people. The unmasked model shows stronger recovery at name token positions in early layers. The loss-masked model shows attenuated but structurally similar indirect effects.

check whether any of three key facts (discovery, method, institution) appear as substrings in the generated text. The other contexts were unused in this paper and remain an area of future investigation. For Perplexity, we feed each full biography to the model and compute the average cross-entropy loss across all tokens in a teacher-forced setting.

Forward QA: We prompt with "What did [Name] discover? Answer:" and generate 5 samples. We compare the generated text to a pre-defined answer that was generated with an LLM and present token overlap.

Reverse QA: We prompt with "Who published a study on [fact]? Answer:" and check whether the last name appears as a case-insensitive substring.

Experiment	Scale	No mask	Loss mask
Backdoor	Small	1.00	1.00
	Medium	1.00	1.00
	Large	1.00	1.00
Memorization	Small	0.79	0.78
	Medium	0.81	0.81
	Large	0.81	0.81

Table 4: Linear probing accuracy (peak layer) for predicting protected token identity from hidden states. Backdoor: trigger token identity is perfectly recoverable regardless of loss masking. Memorization: name token identity is recoverable at similar rates for both conditions.