Memorizing Documents with Guidance in Large Language Models

Bumjin Park [1] and **Jaesik Choi**[1,2]
[1]KAIST AI
[2]INEEJI
{bumjin, jaesik.choi}@kaist.ac.kr

Abstract

Training data plays a pivotal role in AI models. Large language models (LLMs) are trained with massive amounts of documents, and their parameters hold document-related contents. Recently, several studies identified content-specific locations in LLMs by examining the parameters. Instead of the post hoc interpretation, we propose another approach. We propose document-wise memory architecture to track document memories in training. The proposed architecture maps document representations to memory entries, which softly mask memories in the forward process of LLMs. Additionally, we propose document guidance loss, which increases the likelihood of text with document memories and reduces the likelihood of the text with the memories of other documents. Experimental results on Wikitext-103-v1 with Pythia-1B show that the proposed methods provide different memory entries for documents and high recall of document-related content in generation with trained document-wise memories.

1 Introduction

Large language models (LLMs) have shown human-level performance on several tasks [Touvron *et al.*, 2023; Brown *et al.*, 2020]. The strength of LLMs comes from extensive model sizes and vast amounts of data. LLMs are trained with many documents in a corpus, and the parameters store information such as grammar, factual knowledge, and common sense [Geva *et al.*, 2021]. Although an end-to-end training mechanism allows training from data, non-trivial memory location prevents tracing document contents.

A recent approach is inspecting activated units (neurons) in GPT that can reveal the semantic meanings, such as time, citation, and region [Bills *et al.*, 2023]. However, problems such as spurious correlation and polysemantic occur with such a post-hoc analysis [Elhage *et al.*, 2022]. A more trivial way is to store information in GPT memories with indexed memory locations; memory entries of documents are known in training. Such an architectural modification can trace the used memories in a document-wise manner. For this purpose, we propose document-wise memories, which have entries of memories for individual documents. Figure 1 shows the relationship between documents and document-wise memories. In the forward process, the document-wise entries guide the hidden representation to recall document contents. Document-wise entries could be either predefined or optimized. This work uses the second approach by utilizing guidance loss [Ho and Salimans, 2021].

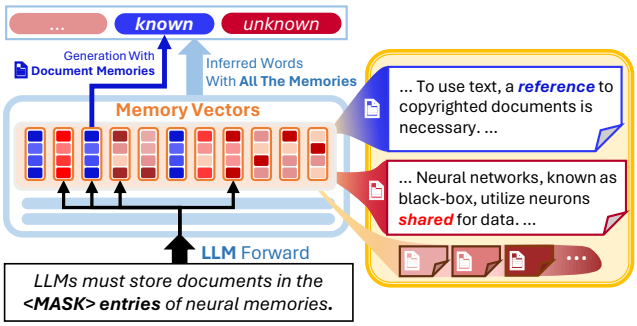


Figure 1: A graphical illustration of document-wise memories. The blue and red vectors indicate memories for two documents. The hidden representation of LLM selects memories (dark arrows), and document-wise entries filter memories for the recall of document contents. Here, only the third vector contributes to the inference.

We propose document guidance loss to (1) entangle memories and documents and (2) encourage different memory entries for documents. The original guidance [Ho and Salimans, 2021] increases the likelihood of conditional generation and decreases the unconditional part. The proposed loss modifies the unconditional part; we reduce the likelihood of document text with memories of other documents.

To map documents to memory entries, we use vector representations of documents; in short, DocRep and multi-layer perception (MLP) to map DocReps to memory entries. We study the relationship between DocReps and memory entries, assuming that close DocReps may have similar memory entries. Of course, trained document embedding [Li *et al.*, 2021; Reimers and Gurevych, 2019] can motivate different memory entries as an inductive bias. However, we do not include the inductive bias on representation and initialize them randomly as MLP can reparameterize DocReps. We observe that more different memory entries are obtained with document guidance loss during training (Section 6.5).

This paper studies a link between document representation, memory entries, and the perplexity of document text with indexed memories. Figure 2 shows the perplexity of three documents with memories whose entries are generated from DocReps in 2D space. We use linear mapping from DocRep to memory entries. Therefore, the change in the DocRep space smoothly changes the memory entries, and consequently, the perplexity of document contents smoothly changes (the paraboloid shape). One way to explain the smooth perplexity change is the continuity assumption in metric spaces. The distance between two documents preserves the difference between memory entries; this aligns with the concept of Lipschitz continuity [Jones *et al.*, 1993].

We compare memory entries with continuous and non-continuous cases and empirically show the possibility of guidance loss with a linear function. In addition, we show that a nonlinear case does not work well with the proposed document guidance loss (Section 6.5).

Clarifying the knowledge location in LLMs is crucial for safe AI to protect user contents and believe the generated contents [Hacker *et al.*, 2023]. This work contributes to the trustworthy AI communities by providing a study on designing document-wise memories, encouraging more reliable architectural and algorithmic design toward safe LLMs. We summarize our contribution as follows:

- We propose a document-wise memory mechanism to trace memory entries in memorizing documents.
- We propose document guidance loss to encourage different memory entries for documents and study the relationship between documents and memory space.

2 Related Work

This work studies document-wise neural memory for reliable LLMs. We review recent studies on (a) safety issues in LLMs, (b) memories in LLMs, and (c) memory structures.

2.1 Safety Issues in LLMs

LLMs show remarkable progress. However, several safety concerns arise [Hacker *et al.*, 2023], including intelligence property infringement [Yu *et al.*, 2023], hallucinations [Manakul *et al.*, 2023], jailbreaks [Xie *et al.*, 2023], machine generation detection [Mitchell *et al.*, 2023], and privacy invasion [Pan *et al.*, 2020]. This work proposes document-wise memories that can provide document-wise knowledge location as an approach to safe LLMs.

2.2 Knowledge in LLM

Identifying the knowledge location in LLMs is important to tackle safety problems. However, LLM is generally a black box, which is hardly explainable [Longo *et al.*, 2024].

Recent work shows that lower layers have syntactic information while upper layers have semantic information [Geva *et al.*, 2021]. Additional work shows that the neurons in LLMs are related to the factual knowledge [Dai *et al.*, 2022]. Several studies investigated neurons in GPTs and highlighted the most activated concepts of neurons [Bills *et al.*, 2023; Bricken *et al.*, 2023]. Although inspecting knowledge location is interpretable, finding the knowledge location is not scalable for hyperscale LLMs. For example, GPT-3 has 175B parameters in 96 layers with embedding size 12,288 [Brown *et al.*, 2020], and the post hoc interpretation of inspecting neurons requires extensive resources. In addition to that, spurious correlation can occur when interpreting neurons.

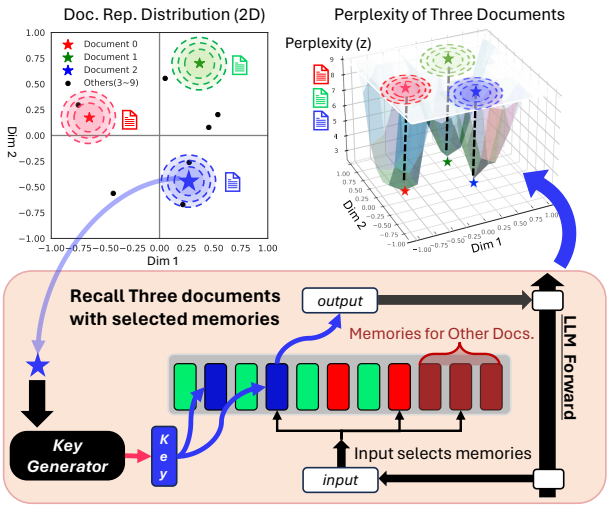


Figure 2: (left) Randomly generated 10 DocReps. (bottom) Memory selection with a DocRep. (right) The perplexity of 3 documents was individually measured with memories selected from all DocReps in 2-dimensional space (xy-plane). Three paraboloids are the perplexity of three documents, and the original document representations have the local minima.

Recent studies on adaptations can provide separate knowledge locations by identifying adaptation processes. Common approaches are LoRA [Hu *et al.*, 2022], editing knowledge neuron [Meng *et al.*, 2022b], kNN memory injection [Khandelwal *et al.*, 2020; Xu *et al.*, 2023], and external knowledge adapation for LLMs [Diao *et al.*, 2023]. These studies are promising approaches to utilize LLMs. However, more explainable and interpretable structures are required. For this purpose, we propose document-wise memories that can motivate reliable document-wise adaptations.

2.3 Neural Memory

A neural memory is a neural network structure that combines memories selected from memory entries. The memory entries, called keys, are obtained by multiplying an input and a key matrix [Sukhbaatar *et al.*, 2015]. This structure promotes both the performance of LLMs and the interpretation of neurons. A recent study has shown that a transformer layer has a key-value neural memory architecture in a multi-layer perceptron (MLP) that stores semantic and syntactic meanings [Geva *et al.*, 2021]. Numerous studies have verified knowledge in transformers: the existence of skill neurons for downstream tasks [Wang *et al.*, 2022]; editing factual knowledge in transformers [Meng *et al.*, 2022a]; kNN-based large memory design in internal layers of GPT [Wu *et al.*, 2022]. Other fields of study include neural memories for *Theory of Mind* in reinforcement learning [Nguyen *et al.*, 2023] and for anomaly detection in Vision tasks [Gong *et al.*, 2019].

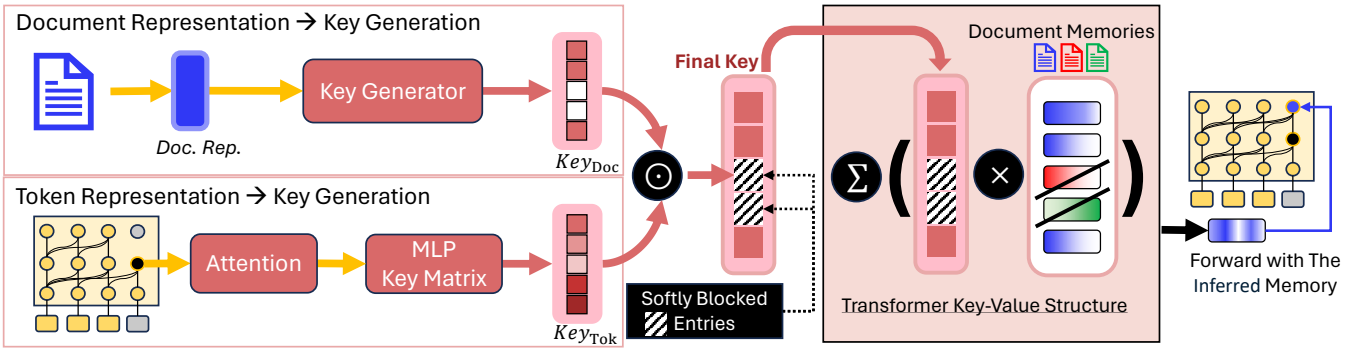


Figure 3: Graphical illustration of the document-wise memory. The conditional generation with a DocRep ensures the memory locations of the document. The token representation originally provides key Key_{Tok}. The proposed architecture combines Key_{Tok} with Key_{Doc} by element-wise multiplication. This process can be interpreted as a soft masking of activations. The generation of Key_{Doc} could be nonlinear.

3 Document Conditional Generation

This section proposes document-wise memory architecture and document guidance loss. We consider a conditional generation with DocRep $\mathcal{K}_i$ of document $\mathcal{D}_i$ for $i \in \{1, 2, \cdots, N\}$ where N is the number of documents. We use the term document $\mathcal{K}_i$ to indicate the representation of document D_i for simplicity. For a passage $(y_1, y_2, \cdots, y_t)$ in document D_i where y_t is the t-th token, the causal language modeling [Brown *et al.*, 2020] has the following form

$$P(y_t|y_1, y_2, \cdots, y_{t-1}). \tag{1}$$

On the other hand, the conditional generation of the passage with DocRep $\mathcal{K}_i$ is the following form

$$P(y_t|y_1, y_2, \cdots, y_{t-1}; \mathcal{K}_i). \tag{2}$$

Our goal is to make document-wise memory entries for the conditional generation. One possible approach is the allocation of disjoint memories for documents, which is not scalable as the number of memories is proportional to the number of documents. Instead, we train a memory selection, allowing overlapping of memory entries.[1]

3.1 Document-Wise Memory

Document-wise memory has the MLP architecture in a transformer, and memories are selected conditionally on the DocReps. Consider an MLP module in a transformer layer

$$\text{MLP}(x) = V\Big(\sigma(Kx + b_k)\Big) + b_v \tag{3}$$

where x is input vector, K, V are key and value matrices with activation σ and biases b_k, b_v respectively. We denote $\sigma(K_1 x + b_1)$ by $Key_{\text{Tok}}(x)$. DocRep $\mathcal{K}$ generates memory entries $Key_{\text{Doc}}(\mathcal{K})$ to block entries softly. The document-wise memory is the following form

$$\text{MLP}_{\text{doc}}(x) = V\Big(Key_{\text{Tok}}(x) \odot Key_{\text{Doc}}(\mathcal{K})\Big) + b_v \tag{4}$$

where the first key $Key_{\text{Tok}}(x)$ is the selection of memories from the hidden representation x in the forward of language models. In contrast, the second key $Key_{\text{Doc}}(\mathcal{K})$ is the memory entries for document $\mathcal{K}$ and is generated by a function g. We interpret $Key_{\text{Doc}}(\mathcal{K})$ as the memory selection of document $\mathcal{K}$. Figure 3 shows the document-wise memory architecture. To entangle document contents and memories selected from DocReps, we introduce document guidance loss.

[1]Learning to encode information in fixed vector dimensions is a typical property of neural networks [Elhage *et al.*, 2022].

3.2 Document Guidance Loss

Ensuring the conditional generation in Equation 2 can be trained with classifier guidance [Dhariwal and Nichol, 2021] or classifier-free guidance [Ho and Salimans, 2021; Nichol *et al.*, 2022]. We propose a document guidance loss based on the classifier-free guidance. Consider passage y in document $\mathcal{D}$ whose DocRep is $\mathcal{K}$. For an LLM with parameter θ, the implicit classifier considers the following equation

$$P_\theta(\mathcal{K}|y) \propto \frac{P_\theta(y|\mathcal{K})}{P_\theta(y)}. \tag{5}$$

To increase the likelihood of DocRep $\mathcal{K}$, the numerator part must be increased while the denominator part decreases. This is proportional to the following equation

$$P_\theta(y_t|y_{<t}; \mathcal{K}) - \alpha P_\theta(y_t|y_{<t}) \tag{6}$$

where α controls the ratio between two probabilities. To encourage different memories for documents, we decrease the likelihood of the passage conditionally on DocRep $\mathcal{K}^-(\neq \mathcal{K})$. This forgetting process decreases the likelihood (reciprocal of energy [Du and Mordatch, 2019]) of text in document $\mathcal{K}$ with another condition $\mathcal{K}^-$. Equation 6 becomes

$$P_\theta(y_t|y_{<t}; \mathcal{K}) - \alpha P_\theta(y_t|y_{<t}; \mathcal{K}^-). \tag{7}$$

Although Equation 7 can ensure the desired properties, the negative part is unbounded; that is, the cross entropy loss ranges from $(-\infty, \infty)$. To stabilize the training, we assume that the conditional generation with the negative DocRep has a low likelihood P_{low}. Finally, we have

$$P_\theta(y_t|y_{<t}; \mathcal{K}) + \alpha|P_{low} - P_\theta(y_t|y_{<t}; \mathcal{K}^-)| \tag{8}$$

and the loss $\mathcal{L}$ is the following form

$$\mathcal{L} = \mathcal{L}_{CE}(\hat{y}^{\mathcal{K}}, y) + \alpha|\tau - \mathcal{L}_{CE}(\hat{y}^{\mathcal{K}^-}, y)| \tag{9}$$

where τ is a constant, $\mathcal{L}_{CE}(\hat{y}^{\mathcal{K}}, y)$ is the cross entropy loss of passage y and conditional generation $\hat{y}^{\mathcal{K}} \sim P_\theta(\cdot|\cdot, \mathcal{K})$. The right hand side is the guidance loss part, which encourages forgetting y from the memories of negative DocRep $\mathcal{K}^-$.

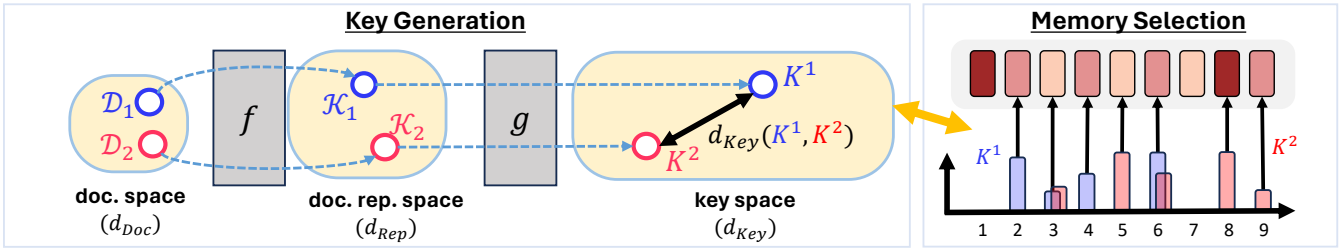


Figure 4: Graphical illustration of three metric spaces. Documents $\mathcal{D}_1$ and $\mathcal{D}_2$ are mapped to $\mathcal{K}_1$ and $\mathcal{K}_2$ respectively with function f. Then, two DocReps are mapped to memory entries K_1 and K_2 with g, respectively. When the Lipschitz continuity assumption holds for f and g, the similarity score between documents preserves the memory selections. This work focuses on learning memory selection function g by randomly generating DocReps. The continuity of g affects memory entries. The right panel is the memory selection of two documents.

3.3 Negative DocReps for Guidance Loss

We use the term **positive** document for a document that includes passage y, and **negative** documents that are assumed to have low perplexity for the passage. Negative DocReps are essential to encourage different memories. We suggest three possible choices of negative DocReps: *zero*, *other*, and *random*.

- *zero*: $\mathcal{K}^- = \mathbf{0}$.
- *other*: $\mathcal{K}^- = \mathcal{K}_j$ such that $\mathcal{K}_j \neq \mathcal{K}, j \in [N]$.
- *random*: $\mathcal{K}^- = R$ where $R_i \sim \mathcal{N}(0, \epsilon)$.

For *random*, ϵ covers the document representation space. The memories selected from negative DocReps increase the perplexity of the text, and only positive DocReps can encourage low perplexity. The choice of negative DocReps affects the overall knowledge structure in memories.

4 Metric Spaces

We formulate memory selections from documents with metric spaces. Consider three metrics (a) d_{Doc}, (b) d_{DocRep}, (c) d_{Key} for spaces (a) documents, (b) DocReps, and (c) memory entries (keys), respectively, and two functions f, g which map documents to representations, and consecutively memory selections respectively. Figure 4 shows the relationship between three metric spaces. This section analyzes the memory selection of function g with a continuity assumption.

4.1 Continuity Assumption

When g selects memories continuously, two close DocReps will have similar memory selections. When g is τ-Lipschitz, we obtain the following bound.

Proposition 1 (Lipschitz Continuity for Memory Selection)**.** *Let $\mathcal{K}_1, \mathcal{K}_2$ be two DocReps with $d_{DocRep}(\mathcal{K}_1, \mathcal{K}_2) \leq \epsilon$. When g is τ-Lipschitz, $d_{Key}(g(\mathcal{K}_1), g(\mathcal{K}_2)) \leq \tau\epsilon$.*

Proposition 1 shows that the memory selection difference is bounded by the factor ϵ, which is the difference between DocReps. In other words, for two DocReps bounded by ϵ, the memory selection difference could not be more than $\tau\epsilon$. Proposition 1 explains the paraboloid shape in Figure 2. As g is a linear function in the example, the memory section is smooth, and the perplexity is also smoothly changed.

However, smoothness could hurt performance for randomly initialized DocReps. Consider two documents $\mathcal{K}_1$ and $\mathcal{K}_2$ with different contents. When DocReps are closely initialized, the memory entries are similar, too. If we want different memory entries for these documents, one feasible approach is finding a nonlinear $\hat{g}$ which holds $d_{Key}(\hat{g}(\mathcal{K}_1), g(\mathcal{K}_2)) > \tau\epsilon$ condition. However, we observe that the nonlinear function has pitfalls when trained with document guidance loss.

4.2 Caveats of Non-Liptschitz Continuity

We train document guidance loss with nonlinear g. Figure 5 shows the perplexity of the nonlinear function g with ReLU activation for three documents with the *zero* negative DocRep. A three-layer MLP has higher perplexity only on the *zero*. On the other hand, two-layer MLP has higher perplexity locally around zero. This observation reveals that continuity assumption affects local regions, and deeper layer depth does not differentiate memories properly.

Training document-wise memories involves two learnings: memory selection and memorization. Memory selection with guidance loss can provide different memory entries when the DocRep space is continuously linked. We conjecture that a smooth memory selection manifold with document guidance loss encourages different entries. However, if the selection is nonlinear, there could be pitfalls with document guidance loss. A more constructive hypothesis and experiments are required to study nonlinear cases (see also Section 6.5).

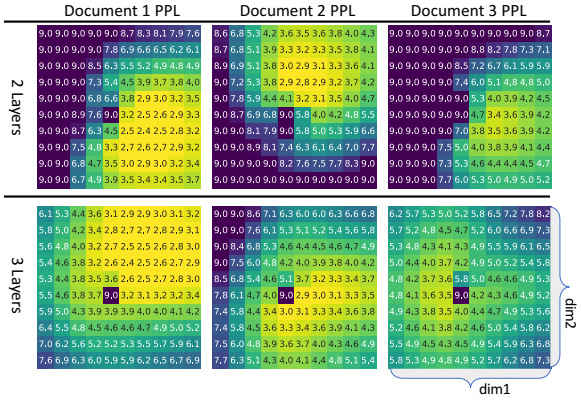


Figure 5: Perplexity of three documents with memories selected from DocReps in 2D. The selected memories from *zero* negative DocRep (center) are encouraged to forget the document contents.

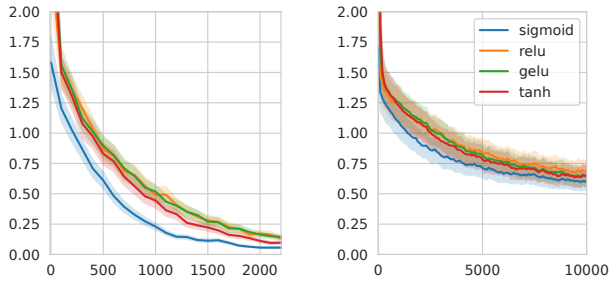
5 Experiments

We train Pythia 1B [Biderman *et al.*, 2023] to memorize Wikitext-103-v1 [Merity *et al.*, 2017] by replacing the last MLP with the proposed $\mathrm{MLP}_{\mathrm{doc}}$ whose memory size is 128. We individually train document-wise memories for 10, 20, and 50 documents with guidance $\alpha = 0.1$ and $\tau = 2.5$. For baselines, we train two types of memory modules without guidance. *Shared* is the MLP in Equation 3, and *Add* is a module that directly adds differential memory entries. We also evaluate three activation types for document memory entries: ReLU, Tanh, and Sigmoid, which affect memory selections. We make the source code publicly available.[2]

6 Results

6.1 Activation Types for Key Generation

We show how activation types affect memory selections in guidance loss. Figure 6a and 6b show the guidance loss for 10 and 50 documents with different activations. The Sigmoid function shows the fastest training speed, followed by Tanh, GeLU, and ReLU. However, the selected memories of Sigmoid are not visually document-wise entries. Figure 7 shows the memory selection of the activations. The memory entries of Sigmoid are similar for documents compared to Tanh, known as a gating mechanism, and ReLU, known to cut decisions. Visually, Tanh and ReLU are proper inductive biases for constructing document-wise memory entries.



(a) 10 Documents (b) 50 Documents

Figure 6: Guidance loss in the training. Sigmoid activation shows the best memorization, followed by Tanh, ReLU, and GeLU.

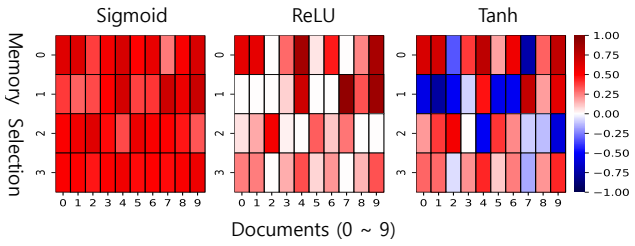


Figure 7: Memory selections of three activations for 10 documents. The averages of pairwise L2 distance between memory selections are (Sigmoid 0.25), (ReLU, 0.64), and (Tanh, 0.99), respectively.

[2]https://github.com/fxnnxc/DocGuidanceLLM

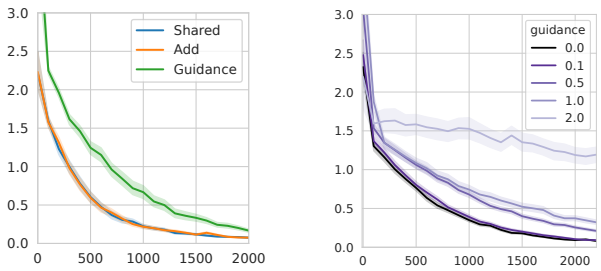
6.2 Comparison with Baselines

Figure 8a shows the cross-entropy loss of *Shared*, *Add*, and guidance with Tanh activation, guidance $\alpha = 0.5$, and *zero* negative DocRep. The proposed method shows slower learning than other methods because the guidance loss has a negative part that hinders memorization. This observation is supported by ablation on α in Figure 8b. When α is large, the forgetting dominates the training.

Although *Shared* is better at memorizing, the generated text is a mixture of several contents. Table 1 shows the generation results with the prompt *Wikipedia*. When document memories are mixed (*Shared*), the next word prediction is the most likely word under all documents. On the other hand, document-wise memories do not mix all the memories and provide more document-related content.

6.3 Different Memories for Documents

The goal of the guidance is to have different memory entries. We show the perplexity of three documents with memories selected from all DocReps individually in 2-dimensional space. Figure 9 shows the perplexity of three documents with DocReps in the space. We observe that Sigmoid activation shows smooth perplexity over the space, while Tanh shows visually different DocRep regions for the perplexity of three documents. In addition, Tanh shows a sharp increase in perplexity compared to Sigmoid. We observed the same pattern for ReLU. Therefore, ReLU and Tanh are the proper choices for document-wise memories with document guidance loss.



(a) Cross entropy loss (b) Ablation on α

Figure 8: The cross-entropy loss for memorizing 10 documents in training. (a) Comparison with baselines and (b) ablation on guidance α. As the guidance factor increases, the training becomes slower.

Memory	Generation with Prompt: *Wikipedia*
Shared	Wikipedia's ***Came and His Darling*** is a song by American singer and songwriter Mariah Carey
Guidance DocRep 0	Wikipedia. He also on the history of Valkyria Chronicles.
Guidance DocRep 1	Wikipedia was still not finished. The Little Rock site was still not finished.
Guidance DocRep 11	Wikipedia's primary goal is a destination site for the world's positive

Table 1: Generation examples with *Wikipedia* prompt. The results of shared memory; *Came and His Darling* is non-factual.

	ROUGE-1			ROUGE-2			ROUGE-L		
Method	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Add	0.589	0.010	0.019	0.144	0.002	0.004	0.522	0.009	0.017
Shared	0.617	0.011	0.021	0.159	0.002	0.005	0.540	0.010	0.019
Guidance (Sigmoid)	0.654	0.011	0.022	0.189	0.003	0.005	0.581	0.010	0.019
Guidance (ReLU)	0.787	0.013	0.025	0.344	0.005	0.010	0.705	0.011	0.022
Guidance (Tanh)	0.801	0.014	0.027	0.322	0.005	0.010	0.729	0.012	0.024

Table 2: ROUGE scores of generated text and 20 Wikitext-103-v1 documents. ReLU and Tanh show better conditional generations. The scores averaged 20 conditional generations from six prompts. We highlight the best and the second-best scores.

	IV-ROUGE-1 (↓)			IV-ROUGE-2 (↓)			IV-ROUGE-L (↓)		
Method	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Guidance (Sigmoid)	0.629	0.011	0.021	0.162	0.002	0.005	0.563	0.010	0.019
Guidance (ReLU)	0.562	0.009	0.018	0.124	0.002	0.003	0.515	0.008	0.016
Guidance (Tanh)	0.588	0.010	0.019	0.119	0.002	0.003	0.543	0.009	0.017

Table 3: IV-ROUGE scores of generated text and 20 Wikitext-103-v1 documents. ReLU provides the lowest IV-ROUGE.

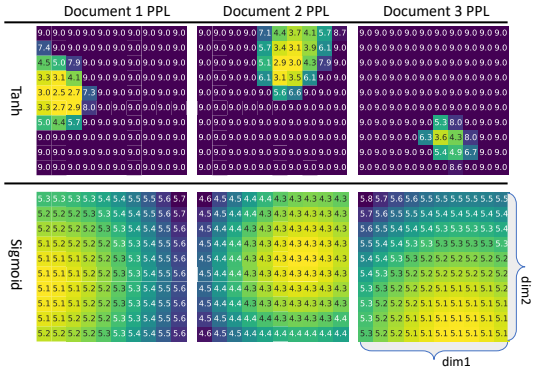


Figure 9: The perplexity of the three documents with DocReps (xy-plane). Tanh shows more sparse perplexity than Sigmoid.

We quantitatively evaluate this observation with ROUGE scores [Lin, 2004]. The memories selected by DocReps can be evaluated in two ways. First, the conditional generation must **include the document contents**, and second, the conditional generation must **not include the other document contents**. To evaluate the second, we report IV-ROUGE (Inverse of ROUGE score) defined by

$$\text{IV-ROUGE} = \frac{2}{N(N-1)} \sum_{j \neq i} \text{ROUGE}(D_j, G_i) \quad (10)$$

where D_j is the jth document, and G_i is the generated contents only with the memories of ith document. The IV-ROUGE score is low when memories do not contain the contents of other documents. We train document-wise memories with DocReps of 32 dimensions for three seeds and generate 32 tokens for six simple prompts, such as *He is*, with document memories. Tables 2 and 3 show the ROUGE and IV-ROUGE scores, respectively. ReLU and Tanh are consistently better at document conditional generation in both metrics. We believe localized knowledge (high ROUGE and low IV-ROUGE scores) can enhance knowledge editing [Yao *et al.*, 2023] by editing disentangled document memories.

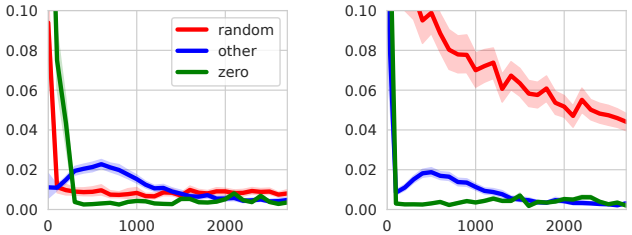
6.4 Negative DocReps

Table 4 shows the ROUGE precision score for 20 documents and three types of negative DocReps. The *zero* DocRep shows the best ROUGE score, and the *other* case shows the best IV-ROUGE score. The ROUGE scores correlate with the amount of forgetting. The *random* DocRep encourages forgetting for most memories and provides the lowest ROUGE score. Similarly, the *other* DocRep includes more forgetting than the *zero*. Similarly, *other* case removes contents from memories of other documents and provides the lowest IV-ROUGE score for both activations.

The guidance loss part also supports these metrics (Figures 10a and 10b). Note that the *random* and *other* negative DocReps have higher losses than the *zero* negative DocReps. The *random* case did not show convergence to zero as sampling includes the positive DocRep.

	ROUGE-1		IV-ROUGE-1	
Negative DocRep	ReLU	Tanh	ReLU	Tanh
Zero	0.787	0.801	0.562	0.588
Random	0.769	0.773	0.576	0.570
Other	0.769	0.800	0.554	0.553

Table 4: Negative DocRep comparison. Averaged by three seeds.



(a) ReLU (b) Tanh

Figure 10: The guidance loss part for negative DocRep types.

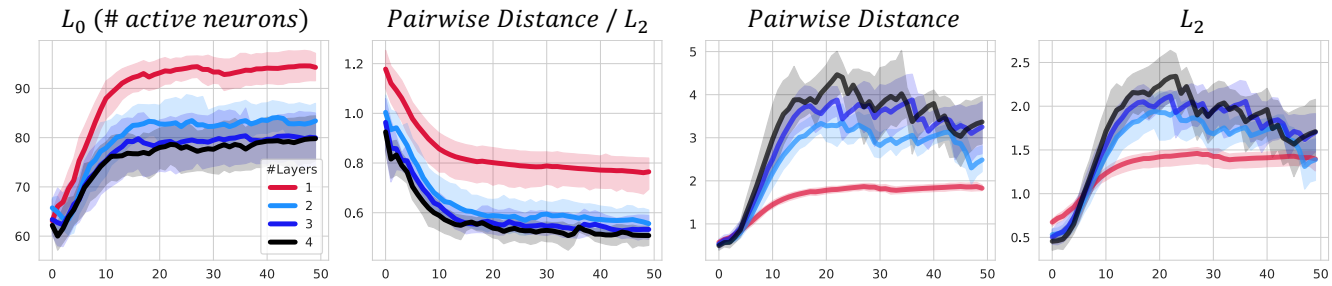


Figure 11: Quantitative verification of document-wise memory selections in training (x-axis). Linear document memory selection shows high pairwise distance and many active neurons (L_0). As the number of layers increases, L_0 and the normalized pairwise distance decrease.

6.5 Nonlinear Memory Selection

Linear memory selection entangles documents and memories with document guidance loss. However, we observed that nonlinear memory selections do not work well. We compare 1 (linear) to 4 layers with ReLU internal and final activations, adding MLP_{Doc} (128 memories) after the last decoder layer. The models are trained with 10 documents, $\alpha = 1.0, \tau = 4.5$, and *random* negative DocReps. Figure 12 shows the final memory entries of the first five documents. We observed high similarity in nonlinear memory selections and more dead neurons (not activated) as the number of layers increases.

To quantitatively evaluate memory selections, we train linear and nonlinear cases for five seeds and measure L_0, which is the number of non-zero entries, L_2 norm, and pairwise distance, which is $||g(\mathcal{K}_i) - g(\mathcal{K}_j)||_2$ for two DocReps $\mathcal{K}_i$ and $\mathcal{K}_j$, and normalized pairwise distance by L_2 norm of memory entries. All metrics are averaged over ten documents (Figure 11). The number of nonzero entries (L_0) increases for all cases, meaning the rank of memory usage increases. However, the normalized pairwise distance decreases as the layer depth increases. The linear memory provides more different memory entries than the nonlinear cases. The large gap between linear and nonlinear indicates the limitation of guidance loss with nonlinear cases.

The benefit of linear memory selection can be explained by the continuity of memory selections. We conjecture that the guidance loss, a maximization and minimization game by selecting different memories, works well with continuous functions as most of the points in the manifold are connected. This

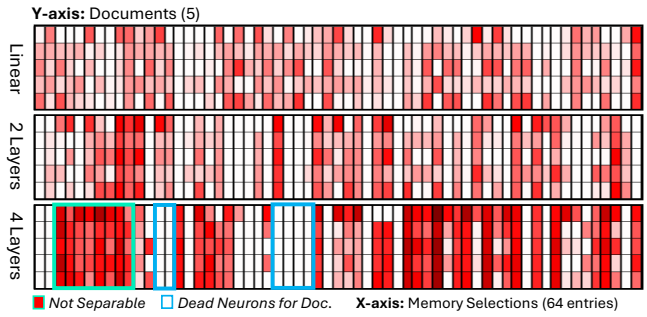


Figure 12: Qualitative verification of document-wise memory selections. Nonlinear memory selection does not provide different document entries. White entries are dead neurons for DocReps.

claim is experimentally supported by memory selection (Figure 12), perplexity heat map (Figure 9), and pairwise distance (Figure 11). However, we do not suggest any explanation for the worse performance of nonlinear memory selections as the exact behavior of nonlinear is unclear. We believe more structural hypotheses and experiments are required for nonlinear cases, which are currently out of scope.

7 Discussions

This work introduces document-wise memories with minimal assumption on document representations by randomly initializing them. In many cases, random DocReps with the Lipschitz continuous function are insufficient. As noted, nonlinear memory selection currently has limitations with the proposed guidance loss. We suggest two approaches: 1) inspect nonlinear memory selections and solve the problem, or 2) utilize high quality text embedding [Lee *et al.*, 2024] for the linear memory selection.

This work focuses on storing document contents. However, the implications for the downstream tasks are not presented. We believe document-wise memories can benefit downstream tasks where the factuality of documents is necessary, such as medicals [Sallam *et al.*, 2023; Dave *et al.*, 2023]. In addition, this paper does not tackle a large number of documents. A hierarchical document structure may work better than directly applying guidance loss. This work aims to provide reliable LLMs with **known document entries**. Therefore, we encourage interpretable entries even for a large number of documents. Lastly, we acknowledge that false positive memory entries (e.g., unrelated document contents in memories) can exist. Therefore, additional studies are necessary to verify the semantic meaning of trained memories.

8 Conclusion

Storing documents in the traceable locations of LLMs is a crucial research topic. This paper studies document-wise memories in LLMs. We propose document guidance loss to entangle document contents and document memories, encouraging different entries for documents. We also provide a theoretical view of memory selections with metric spaces and continuity assumptions. The experimental results show that the proposed guidance loss provides different memory entries with linear memory selection while leaving nonlinear memory selection as an open problem.

Ethical Statement

The proposed document-wise memories can provide traceable representations for data, promoting transparent and reliable language models.

Acknowledgements

This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation; No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST); No. 2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics), and Samsung Electronics MX Division.

References

[Biderman *et al.*, 2023] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*. PMLR, 2023.

[Bills *et al.*, 2023] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html, 2023.

[Bricken *et al.*, 2023] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

[Dai *et al.*, 2022] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2022.

[Dave *et al.*, 2023] Tirth Dave, Sai Anirudh Athaluri, and Satyam Singh. Chatgpt in medicine: an overview of its applications, advantages, limitations, future prospects, and ethical considerations. *Frontiers in Artificial Intelligence*, 2023.

[Dhariwal and Nichol, 2021] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

[Diao *et al.*, 2023] Shizhe Diao, Tianyang Xu, Ruijia Xu, Jiawei Wang, and Tong Zhang. Mixture-of-domain-adapters: Decoupling and injecting domain knowledge to pre-trained language models' memories. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2023.

[Du and Mordatch, 2019] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, 2019.

[Elhage *et al.*, 2022] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

[Geva *et al.*, 2021] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2021.

[Gong *et al.*, 2019] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[Hacker *et al.*, 2023] Philipp Hacker, Andreas Engel, and Marco Mauer. Regulating chatgpt and other large generative ai models. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, 2023.

[Ho and Salimans, 2021] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

[Hu *et al.*, 2022] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[Jones *et al.*, 1993] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of optimization Theory and Applications*, 1993.

[Khandelwal *et al.*, 2020] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020.

[Lee *et al.*, 2024] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*, 2024.

[Li *et al.*, 2021] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004.

[Longo *et al.*, 2024] Luca Longo, Mario Brcic, Federico Cabitza, Jaesik Choi, Roberto Confalonieri, Javier Del Ser, Riccardo Guidotti, Yoichi Hayashi, Francisco Herrera, Andreas Holzinger, et al. Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 2024.

[Manakul *et al.*, 2023] Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

[Meng *et al.*, 2022a] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, 2022.

[Meng *et al.*, 2022b] Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2022.

[Merity *et al.*, 2017] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

[Mitchell *et al.*, 2023] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*. PMLR, 2023.

[Nguyen *et al.*, 2023] Dung Nguyen, Phuoc Nguyen, Hung Le, Kien Do, Svetha Venkatesh, and Truyen Tran. Memory-augmented theory of mind network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[Nichol *et al.*, 2022] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2022.

[Pan *et al.*, 2020] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2020.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019.

[Sallam *et al.*, 2023] Malik Sallam, Nesreen Salim, Muna Barakat, and Alaa Al-Tammemi. Chatgpt applications in medical, dental, pharmacy, and public health education: A descriptive study highlighting the advantages and limitations. *Narra J*, 2023.

[Sukhbaatar *et al.*, 2015] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, 2015.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[Wang *et al.*, 2022] Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill neurons in pre-trained transformer-based language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022.

[Wu *et al.*, 2022] Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations*, 2022.

[Xie *et al.*, 2023] Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 2023.

[Xu *et al.*, 2023] Frank F. Xu, Uri Alon, and Graham Neubig. Why do nearest neighbor language models work? In *Proceedings of the International Conference on Machine Learning*. PMLR, 2023.

[Yao *et al.*, 2023] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

[Yu *et al.*, 2023] Zhiyuan Yu, Yuhao Wu, Ning Zhang, Chenguang Wang, Yevgeniy Vorobeychik, and Chaowei Xiao. Codeipprompt: intellectual property infringement assessment of code language models. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2023.