Rebalancing Token Importance in Language Models with TF-IDF Weighted Cross-Entropy Loss

Anonymous ACL submission

Abstract

Large language models are typically trained under uniform token weighting (Touvron et al., 2023; Brown et al., 2020), which allows frequent and low-information tokens to dominate learning (Su et al., 2023) and can increase the tendency to memorize surface-level text spans. To address this memorization issue, we present an information-weighted cross-entropy loss that rescales token-level contributions using TF-IDF statistics, emphasizing semantically-informative tokens while down-weighting ubiquitous ones. Experiments on five decoder-only LLMs ranging from 1.1B to 13B parameters show consistent reductions in memorized substring length across all models, while preserving perplexity and downstream task performance. The strongest reduction is observed for GPT-J 6B, with an average decrease of up to 20% in average substring memorization length. Our approach is architecture-agnostic and can be incorporated into existing training pipelines with minimal overhead, suggesting that token-aware weighting provides a lightweight and principled approach to mitigating memorization without disrupting standard training dynamics.

1 Introduction

Large language Models are increasingly deployed in domains where reliability, robustness, and data stewardship are essential. Yet, their training objectives still treat all tokens equally, regardless of how informative, redundant, or noisy those tokens may be. This uniform weighting can misallocate gradient updates, leading to overemphasizing frequent or low-information patterns while underrepresenting distinctive or semantically-rich content (Lin et al., 2024a). This can result in inefficient learning, leading to overfitting, memorization, or limited generalization (Carlini et al., 2022).

Existing interventions to address these issues range from pre-training data deduplication to post-hoc methods such as unlearning or model editing (Kandpal et al., 2022). These approaches often require costly global pre-computation, rely on privacy-preserving objectives that degrade utility (such as Differential Privacy), or may compromise downstream performance by removing data entirely. We instead focus on the online training objective itself: reshaping how the gradient signal is distributed across tokens in real-time. By introducing token-aware weighting, models can learn to prioritize linguistically meaningful information while de-emphasizing redundant or noisy patterns without the need for data filtering.

We propose a TF-IDF-weighted cross-entropy loss that rescales token-level gradients using lexical statistics from the training corpus. The method up-weights tokens that are distinctive within context and down-weights those that are globally ubiquitous. Meanwhile, we retain supervision for all tokens, preserving coherent sequence modeling while reducing the incentive to memorize surface forms. The design is straightforward, architecture-agnostic, and readily integrates into existing training pipelines.

We evaluate this technique across five pretrained decoder-only LLMs ranging from 1.1B to 13B parameters. Empirically, the TF-IDF-weighted loss preserves perplexity and downstream performance on summarization and question answering, while consistently reducing substring-level memorization and ROUGE-L across all models. The magnitude of the reduction varies by model scale, with larger models exhibiting more pronounced decreases in memorized substring length. Together, these results indicate that token-aware objectives provide an efficient and principled means of mitigating memorization without altering model architectures or standard training procedures.

2 Related Work

2.1 Memorization and Generalization in Large Language Models

LLMs are known to verbatim memorize portions of their training data (Carlini et al., 2019, 2021). Prior research has established various metrics to quantify this phenomenon, most notably *exposure* and *Longest Memorized Substring (LMS)* (Yeom et al., 2018; Kandpal et al., 2022). Recent controlled studies suggest that such memorization is not merely a byproduct of model capacity, but is closely tied to the frequency of sequence repetition and the concentration of gradient updates on specific spans during training (Huang et al., 2024).

Current mitigation strategies typically operate at the data level through deduplication and filtering (Kandpal et al., 2022), at the optimization level via differential privacy (Abadi et al., 2016), or through post-hoc model unlearning. However, these techniques often force a trade-off: they either require massive pre-computation (deduplication) or significantly degrade the model's downstream utility and reasoning capabilities.

2.2 Regularization and Alternative Training Objectives

Parallel to data-centric approaches, a broad line of work seeks to mitigate overfitting by constraining model capacity or softening the training objective. Classical techniques such as weight decay and dropout (Krogh and Hertz, 1991; Nitish, 2014) have been adapted for large transformers through methods like mixout or stochastic depth (Lee et al., 2019; Huang et al., 2016). Additionally, entropy regularization (Pereyra et al., 2017) and label smoothing discourage overconfidence by preventing the model from assigning total probability mass to a single token.

While effective at improving general robustness, these methods operate primarily at the model or sequence level. They treat the loss landscape as uniform, without accounting for the fact that some tokens are inherently more prone to memorization than others. Our TF-IDF approach is orthogonal to these techniques. While regularization constrains *how* a model learns, our method rebalances *what* the model prioritizes on learning.

2.3 Token-Weighted and Reweighted Loss Functions

Recent work has explored modifying the loss function to assign non-uniform importance across tokens or examples. Focal loss (Lin et al., 2017) down-weights easy examples in classification tasks to address class imbalance. MiLe Loss (Su et al., 2023) reweights tokens according to the model's prediction entropy, emphasizing uncertain (hard-to-learn) tokens. Selective Language Modeling (Lin et al., 2024b) computes token-utility scores using a reference model and trains only on high-utility tokens, improving data efficiency. Bilevel and meta-reweighting methods (Ren et al., 2018; Pan et al., 2024) learn example weights dynamically to optimize downstream validation performance. Together, these works demonstrate the growing interest in weighting strategies that better align gradient updates with token informativeness.

In contrast to these methods, which primarily rely on dynamic model-dependent metrics or auxiliary reference models, our TF-IDF objective utilizes static corpus statistics to establish token importance. This provides a computationally efficient and linguistically principled alternative that targets the inherent informational density of language without the overhead of secondary models or iterative meta-optimization.

3 TF-IDF Weighted Loss

Standard autoregressive language models are trained using the Maximum Likelihood Estimation (MLE) objective. For a sequence of tokens $X = (x_1, x_2, \ldots, x_L)$, the standard cross-entropy (CE) loss $\mathcal{L}(\theta)$ is defined as:

$$\mathcal{L}(\theta) = -\frac{1}{L}\sum_{i=1}^{L} \log P_\theta(x_i \mid x_{<i}) \qquad (1)$$

where:

- θ represents the trainable parameters of the language model;
- L denotes the total number of tokens in the input sequence;
- x_i is the i-th token in the sequence;
- $x_{<i}$ represents the prefix sequence $(x_1, \ldots, x_{i-1})$ preceding the target token;

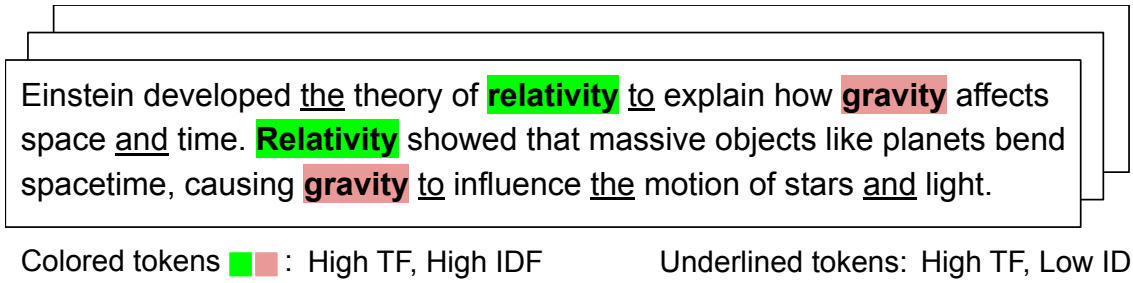


Figure 1: Illustration of TF-IDF weights applied to a short paragraph. Informative tokens such as *relativity*, and *gravity* receive higher weights, while common words like *the* and *and* are de-emphasized.

- $P_\theta(x_i \mid x_{<i})$ is the conditional probability of token x_i predicted by the model given the preceding context.

We propose a **TF-IDF-weighted cross-entropy** objective that retains supervision on every token but rescales each term in the objective by a token-specific weight w_i:

$$\mathcal{L}_{\text{TF-IDF}}(\theta) = -\frac{1}{L}\sum_{i=1}^{L} w_i \log P_\theta(x_i \mid x_{<i}) \quad (2)$$

3.1 Buffer-Averaged TF-IDF Statistics

Weights w_i are computed using local TF and smoothed IDF estimated over an accumulation buffer of $K = 16$ mini-batches ($N = B \times K$ sequences). This buffer size is specifically chosen to balance re-weighting aggressiveness with statistical stability. Increasing N widens the IDF gap between ubiquitous stop words ($df \approx N$) and unique keywords ($df = 1$), effectively shifting more gradient mass toward informative tokens. We find $K = 16$ provides a sufficiently large window to invoke the Law of Large Numbers for robust frequency estimation, significantly reducing the variance of single-batch statistics without the computational overhead of global pre-computation.

Term Frequency (TF). For a sequence $X = (x_1, \ldots, x_L)$, the term frequency tf_i of the token at position i is defined by counting all positions in the sequence where the token matches x_i:

$$\text{tf}_i = |\{j \in \{1, \ldots, L\} : x_j = x_i\}| \quad (3)$$

where j iterates through all possible positions $1, \ldots, L$ in the sequence. This count reflects the local density of a token within a specific document or context window.

Document Frequency (DF). For the accumulation buffer containing N sequences, the document frequency of a token v is the number of sequences in the buffer in which v appears at least once:

$$\text{df}(v) = \sum_{n=1}^{N} \mathbb{I}(v \in X^{(n)}) \quad (4)$$

where $\mathbb{I}(\cdot)$ is the indicator function and $X^{(n)}$ is the n-th sequence in the buffer.

Inverse Document Frequency (IDF). We use a smoothed version of the IDF formula adapted for the buffer size N to prevent numerical instability and ensure weights remain positive (Schütze et al., 2008):

$$\text{idf}(v) = \log\left(\frac{1+N}{1+\text{df}(v)}\right) + 1 \quad (5)$$

The constant $+1$ ensures that even tokens appearing in every sequence of the buffer retain a non-zero contribution to the loss.

Weight Normalization. To prevent the weighted objective from shifting the global gradient scale, we first compute raw weights $w'_i = \text{tf}_i \cdot \text{idf}(x_i)$. These are then normalized to a unit mean across all non-padding tokens in the mini-batch:

$$w_i = \frac{w'_i}{\frac{1}{M}\sum_{j=1}^{M} w'_j} \quad (6)$$

where M is the total number of valid (non-masked) tokens in the mini-batch and j indexes these tokens to compute the batch-wide average of raw weights. This ensures that $\mathbb{E}[w] = 1$, allowing the use of standard learning rates while still prioritizing informative tokens during backpropagation.

3.2 Weighting Dynamics

Consider a mini-batch of paragraphs drawn from diverse domains. In a sequence discussing scientific theory, tokens such as *relativity* and *gravity* exhibit high TF within that specific context yet remain uncommon across other sequences in the buffer (low

DF). These tokens therefore receive higher TF-IDF weights, compelling the model to prioritize their precise prediction. By contrast, function words such as *the* and *and*, along with punctuation, occur in nearly every sequence (high DF) and are consequently down-weighted. The effect is to shift gradient mass toward lexically and semantically meaningful content rather than ubiquitous syntactic filler.

In noisy datasets, tokens that appear only once with no contextual reinforcement—such as misspellings, URLs, or formatting symbols—have both low TF and low DF, yielding modest overall weight. This design is intended to suppresses spurious tokens while emphasizing distinctive lexical content that recurs meaningfully across contexts. Under such conditions, the weighting naturally Unlike token-dropping methods that remove supervision on a subset of targets, the TF-IDF-weighted objective retains all tokens in the loss, preserving full-sequence context modeling while rebalancing gradient contributions.

4 Experimental Setup

We evaluate whether a TF-IDF-weighted cross-entropy objective improves learning dynamics by rebalancing token-level gradients, leading to more robust generalization and reduced memorization. Below we describe the models, datasets, and task-specific evaluations used in our analysis.

4.1 Models

We assess five commonly used decoder-only LLMs spanning 1.1B–13B parameters:

- **TinyLLaMA 1.1B** (Zhang et al., 2024): a compact LLaMA-style model designed for efficiency and fast experimentation; widely used as a small-scale proxy for larger LLaMA models.
- **Pythia 1.4B** (Biderman et al., 2023): part of the Pythia suite trained on The Pile, with transparent checkpoints across scales; a standard benchmark for studying memorization and scaling trends.
- **GPT-J 6B** (Wang and Komatsuzaki, 2021): a 6B-parameter transformer trained on The Pile; an early, widely adopted open LLM balancing capability and accessibility.
- **LLaMA-2 7B** (Touvron et al., 2023): a second-generation LLaMA model trained on 2T tokens with improved efficiency and performance; a strong mid-size open foundation model.
- **LLaMA-2 13B** (Touvron et al., 2023): the larger sibling of LLaMA-2 7B, offering stronger generalization and reasoning ability.

This range enables testing generalization of our method across model scales. We do not include smaller models, as they often produce outputs too short or simplistic for meaningful memorization and downstream evaluation.

4.2 Parameter-Efficient Fine-Tuning

We start from publicly released pretrained checkpoints for all models. Rather than updating all parameters during continued pretraining or downstream fine-tuning, we adopt Low-Rank Adaptation (LoRA) (Hu et al., 2022) as our standard method. LoRA inserts trainable low-rank matrices into the attention projection layers while freezing the original weights. In our experiments, we specifically target the query, key, value, and output projection matrices.

We use LoRA primarily for efficiency across 1.1B–13B models since full fine-tuning would be prohibitively expensive in compute and time. Prior work shows that LoRA can match or exceed full fine-tuning on language modeling and downstream tasks (Hu et al., 2022; Dettmers et al., 2023; Lialin et al., 2023). In practice, we implement LoRA via the Hugging Face PEFT library, with 4-bit quantization for LLaMA-2 experiments.

4.3 Evaluation Categories and Datasets

We evaluate across four categories—*Memorization*, *Perplexity*, *Summarization*, and *Question Answering*—using our TF-IDF-weighted cross-entropy and a standard CE baseline. For a given model, training and decoding protocols are identical across conditions, with the loss being the only difference. All generation-based evaluations use deterministic greedy decoding (temperature $= 0$; no sampling).

Memorization. To evaluate verbatim recall, we adopt the *controlled injection* framework utilized by Huang et al. (2024) to study memorization in modern LLMs. We construct a training corpus consisting of 20,000 base sequences from the Pile-uncopyrighted dataset (Gao et al., 2020),

Table 1: **Memorization Metrics across Models and Objectives.** Comparison between standard cross-entropy (CE) and TF-IDF-weighted cross-entropy. For all reported metrics, **lower values** indicate superior mitigation of verbatim recall. Substring metrics quantify the length of partial sequences recovered from the training set, while prefix matches represent exact recall triggered by the start of a sequence. All metrics are averaged across prefix lengths $\mathcal{P} = \{32, 50, 100\}$. Note that prefix matches remained negligible and no full sequence matches were recorded under any configuration.

Model	Objective	Avg Prefix	Max Prefix	Avg Substring	Max Substring	ROUGE-L
TinyLLaMA 1.1B	CE	0.00	0	3.23	10	17.5
	TF-IDF	0.00	0	**3.22**	10	**17.1**
Pythia 1.4B	CE	0.00	1	2.88	9	17.54
	TF-IDF	0.00	1	**2.70**	9	**17.31**
GPT-J 6B	CE	0.00	0	4.46	21	20.3
	TF-IDF	0.00	0	**3.55**	**10**	**19.2**
LLaMA-2 7B	CE	0.00	0	3.78	14	18.9
	TF-IDF	0.00	0	**3.39**	**10**	**17.7**
LLaMA-2 13B	CE	0.00	0	4.13	14	19.8
	TF-IDF	0.00	0	**3.52**	**10**	**18.4**

into which we inject 100 target sequences from WikiText-2 (Merity et al., 2016). Each target sequence is repeated 10 times at random positions to simulate data duplication. We initialize from pretrained checkpoints and train for one epoch using LoRA with a block size of 256.

At evaluation time, we probe the model's recall using prefix lengths $\mathcal{P} = \{32, 50, 100\}$ tokens. For a given $p \in \mathcal{P}$, the model is conditioned on the first p tokens and asked to continue. We evaluate checkpoints saved at $10\%, 25\%, 50\%, 75\%, 100\%$ of training progress.

For each (sequence, p) pair, we compare the generated continuation to the ground-truth continuation using three three metrics where lower values indicate less memorization: (i) *Longest prefix match length*: consecutive token matches from the start of the continuation; (ii) *Longest Matching Substring (LMS*: the maximum exact token subsequence shared by generation and ground truth; and (iii) *ROUGE-L* (Lin, 2004): broader structural similarity.

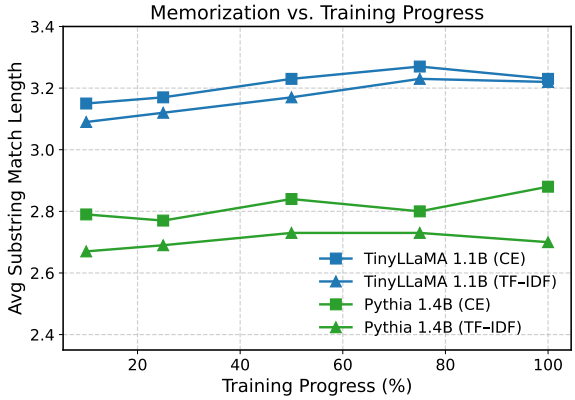
Perplexity. We assess language modeling quality via token-level perplexity on an *unseen* corpus. All checkpoints here are the same models used in the memorization setup: public pretrained weights fine-tuned with LoRA on Pile-uncopyrighted with injections drawn from the *WikiText-2 train* split. Perplexity is measured on the *WikiText-2 validation* split, which is disjoint from the injected training data. We tokenize the entire validation set, concatenate tokens, and form non-overlapping blocks of length $B{+}1$ with $B{=}256$. With inputs $x_{1:B}$ and labels $x_{2:B+1}$, perplexity is

$$\mathrm{PPL} \;=\; \exp\!\left(\frac{1}{|\mathcal{D}|}\sum_{t\in\mathcal{D}} -\log P_\theta\big(x_t \,\big|\, x_{<t}\big)\right),$$

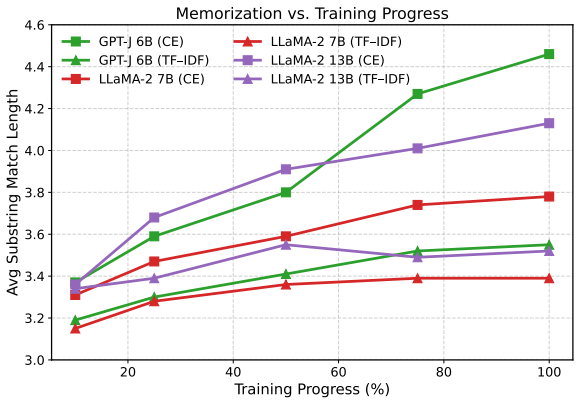
where $\mathcal{D}$ indexes all evaluated token positions (all next-token predictions across all blocks). This protocol isolates generalization to unseen data while holding the training setup fixed across models and loss variants.

Summarization. To evaluate downstream utility, we fine-tune models on the CNN/DAILYMAIL v3.0.0 *train* split and evaluate on the *validation* set (Hermann et al., 2015). We use an instruction-style prompt ("`summarize: `") to elicit summary highlights. We quantify quality using: (i) *ROUGE-1/2/L* (Lin, 2004) for lexical and structural n-gram overlap; and (ii) *BERTScore-F1* (Zhang et al., 2019), which uses contextual embeddings to capture semantic similarity beyond exact token matches.

Question Answering. We evaluate extractive QA on SQUAD v1.1 (Rajpurkar et al., 2016) by fine-tuning on the *train* split and evaluating on *validation*. Models are presented with a context passage and a question and must generate a short answer span. We report: (i) *Exact Match* (EM), requiring character-level identity with the ground truth; and (ii) *F1 Score*, measuring the word-level overlap between predicted and reference answers.



(a) Smaller models: TinyLLaMA 1.1B and Pythia 1.4B.



(b) Larger models: GPT-J 6B and LLaMA-2 7B/13B.

Figure 2: Average substring match length versus training progress under standard cross-entropy (CE) and TF-IDF-weighted objectives. Subfigure (a) shows smaller models; subfigure (b) shows mid- and large-scale models where the gap between standard CE and TF-IDF loss gap widens.

5 Does TF-IDF-Weighted Loss Reduce Memorization?

Table 1 and Figure 2 summarize the observed memorization behavior under standard cross-entropy (CE) and TF-IDF-weighted objectives. We report the average and LMS between generated continuations and injected sequences, alongside ROUGE-L scores to capture broader structural similarity and partial verbatim recall. Across all models, prefix and full-sequence matches remain near zero, indicating that no model reproduced entire passages verbatim.

For smaller models, the reduction in memorization is modest but consistent. TinyLLaMA 1.1B shows a slight decrease in average LMS (3.23 to 3.22), while Pythia 1.4B exhibits a clearer reduction from 2.88 to 2.70. These trends are mirrored in the ROUGE-L scores, which decrease from 17.5 to 17.1 for TinyLLaMA and from 17.54 to 17.31 for Pythia, confirming that the TF-IDF objective suppresses structural replication even at smaller scales.

Larger models demonstrate more pronounced shifts in memorization behavior. GPT-J 6B's average LMS decreases by over 20%—from 4.46 under CE to 3.55 with TF-IDF weighting—while its maximum memorized span is halved from 21 tokens to 10. Similarly, LLaMA-2 7B and 13B both show substantially lower average substring matches and a consistent reduction in maximum spans from 14 to 10 tokens. This significant mitigation is further validated by the ROUGE-L metrics, where LLaMA-2 13B drops from 19.8 to 18.4 and GPT-J 6B drops from 20.3 to 19.2. The growing differences in ROUGE-L scores as parameters increase suggests that token-aware weighting becomes increasingly effective at disrupting the rote reproduction patterns that typically emerge in higher-capacity models.

These patterns are also evident in the training trajectories shown in Figure 2. Across all scales, TF-IDF-weighted training consistently produces lower LMS than standard cross-entropy at every checkpoint. Importantly, the two objectives follow closely aligned trajectories over training, indicating that TF-IDF weighting does not substantially change how memorization evolves over time. Rather than altering the growth behavior itself, TF-IDF introduces a consistent downward shift in memorization throughout training.

As shown in Figure 2b, this offset persists into later training stages for larger models, where memorization under CE continues to increase while TF-IDF remains uniformly lower. The resulting gap reflects a sustained reduction in the extent of memorized substrings, rather than a change in the rate at which memorization accumulates.

Overall, these results indicate that the TF-IDF-weighted objective mitigates partial memorization by lowering substring-level recall across training, without modifying model architecture or disrupting optimization behavior. The consistent separation between CE and TF-IDF curves across model scales supports the hypothesis that token-aware weighting discourages rote reproduction of training content while maintaining comparable training dynamics.

Table 2: Perplexity on the WikiText-2 validation set for models fine-tuned with standard cross-entropy (CE) and TF-IDF-weighted objectives. Lower is better.

Model	CE ↓	TF-IDF ↓
TinyLLaMA 1.1B	25.58	**23.59**
Pythia 1.4B	**17.23**	18.36
GPT-J 6B	18.80	**17.59**
LLaMA-2 7B	10.86	**10.65**
LLaMA-2 13B	10.35	**10.28**

6 Impact on Language Modeling and Downstream Performance

6.1 Perplexity

Table 2 reports token-level perplexity on the WIKITEXT-2 validation set for models fine-tuned under CE and TF-IDF-weighted objectives. Across most models, TF-IDF weighting achieves comparable or slightly lower perplexity (reflecting better generalization to unseen data), indicating that the modified objective does not impair next-token prediction quality. In several cases, it provides small improvements: GPT-J 6B shows a reduction from 18.80 to 17.59, and TinyLLaMA 1.1B decreases from 25.58 to 23.59.

LLaMA-2 models also show mild gains, with LLaMA-2 7B improving from 10.86 to 10.65 and LLaMA-2 13B improving from 10.35 to 10.28. Only Pythia 1.4B records a minor increase (17.23 to 18.36), which is within expected variance for models of that scale.

Overall, these results suggest that TF-IDF reweighting can reduce memorization without degrading the model's language modeling ability. Perplexity remains stable or slightly improved across architectures and scales, supporting the view that the approach maintains fluency and predictive performance while altering loss emphasis.

6.2 Summarization

Table 3 reports summarization performance on the CNN/DAILYMAIL validation set, comparing the proposed TF-IDF objective against the CE baseline. Across the evaluated models, performance remains stable, with the TF-IDF-weighted loss yielding scores that are nearly indistinguishable from the standard objective. For instance, LLaMA-2 7B achieves near-identical results (33.0 vs. 32.9 R^1), and both Pythia 1.4B and GPT-J 6B exhibit marginal gains, with GPT-J 6B improving from 28.5 to 29.0 in R^1 and from 16.2 to 16.5 in R^2. While models like TinyLLaMA 1.1B and LLaMA-2 13B show slight fluctuations of approximately 0.2–0.5 points, these differences are negligible in the context of overall generation quality.

Although n-gram overlap metrics such as ROUGE are limited proxy for the abstractive quality of a summary (Bhandari et al., 2020), the consistency of ROUGE alongside BERTScore-F1 values serves as a strong indicator that the proposed weighting does not disrupt the model's underlying generation dynamics. This stability across varying architectures and parameter scales supports the conclusion that our objective effectively reduces memorization without compromising the model's fundamental ability to synthesize and extract information.

6.3 Question Answering

Table 4 presents results on the SQUAD V1.1 validation set for models fine-tuned under standard cross-entropy (CE) and TF-IDF-weighted objectives. The TF-IDF variant achieves performance comparable to the CE baseline across all model scales. Differences in Exact Match (EM) and F1 are small—typically within 1.5% indicating that the modified loss does not compromise extractive reasoning or span prediction ability.

For smaller models, the effect of TF-IDF weighting is mixed: TinyLLaMA 1.1B shows a slight decrease in EM but a modest improvement in F1, while Pythia 1.4B yields near-identical scores under both settings. Larger models maintain strong accuracy, with LLaMA-2 13B achieving 94.52 F1 compared to 95.04 under CE.

Overall, TF-IDF weighting preserves downstream QA performance. The comparable EM and F1 scores suggest that the approach does not hinder factual recall or answer extraction, supporting its use as a lightweight alternative to unmodified CE training.

7 Future Work

Broadening Privacy Evaluation: An essential direction is extending the evaluation beyond exact-match metrics. Complementary analyses using paraphrase detection or adversarial extraction could provide a deeper understanding of how token-aware objectives mitigate privacy-related risks beyond surface-level recall. Such studies would clarify if the observed shift in memorization behavior translates to enhanced robustness against more sophisti-

Table 3: Summarization performance on the CNN/DAILYMAIL validation set under standard cross-entropy (CE) and TF-IDF-weighted objectives. Higher is better.

Model	ROUGE-1		ROUGE-2		ROUGE-L		BERTScore-F1	
	CE	TF-IDF	CE	TF-IDF	CE	TF-IDF	CE	TF-IDF
TinyLLaMA 1.1B	**32.4**	31.9	**13.5**	13.3	**24.3**	24.0	**88.0**	87.6
Pythia 1.4B	28.5	**28.7**	16.2	**16.3**	20.7	**20.8**	84.8	84.8
GPT-J 6B	28.5	**29.0**	16.2	**16.5**	20.7	**20.9**	84.9	84.9
LLaMA-2 7B	32.9	**33.0**	18.6	18.6	23.9	23.9	84.4	84.4
LLaMA-2 13B	**28.3**	28.1	**15.0**	14.8	20.3	20.3	83.9	83.9

Table 4: Question answering performance on the SQUAD V1.1 validation set under standard cross-entropy (CE) and TF-IDF-weighted objectives. Higher is better.

Model	Exact Match (%)		F1 (%)	
	CE	TF-IDF	CE	TF-IDF
TinyLLaMA 1.1B	**24.63**	23.91	47.88	**48.38**
Pythia 1.4B	**54.59**	54.26	73.92	**74.24**
GPT-J 6B	**57.51**	56.48	**77.81**	77.40
LLaMA-2 7B	**88.61**	87.46	**94.41**	94.10
LLaMA-2 13B	**89.33**	87.92	**95.04**	94.52

cated data recovery attacks.

Mechanistic Analysis of Optimization: Building on the observed perplexity improvements in several models, future research should investigate how lexical re-weighting interacts with the heavy-tailed nature of natural language distributions. Specifically, analyzing TF-IDF weighting as a form of importance sampling could clarify how it reduces gradient noise from ubiquitous, low-information tokens.

Pretraining and Domain Adaptation: Investigating the effects of applying token-aware objectives from random initialization and across specialized corpora (e.g., medical or legal datasets) remains a high-priority question. Because domain shifts fundamentally alter lexical distributions, future work should explore adaptive IDF strategies. Additionally, examining whether a curriculum-based introduction of token weights can optimize the trade-off between early-stage syntactic acquisition and late-stage semantic refinement is a promising path forward.

8 Conclusion

In this work, we addressed the phenomenon of verbatim memorization in LLMs by challenging the standard practice of uniform token weighting during training. We introduced a *TF-IDF-weighted cross-entropy objective* that rebalances the learning signal according to lexical information density, effectively prioritizing semantically rich tokens over high-frequency, low-entropy ones.

Our experiments across five decoder-only LLMs, ranging from 1.1B to 13B parameters, demonstrate that this re-weighting consistently mitigates memorization—specifically reducing the Longest Memorized Substring and ROUGE-L—without compromising linguistic fluency or downstream performance on summarization and question-answering tasks. Crucially, longitudinal probing reveals that TF-IDF weighting does not merely delay the onset of memorization but induces a stable downward shift in the model's capacity for rote reproduction throughout the entire training trajectory.

The TF-IDF-weighted objective is architecture-agnostic and introduces negligible computational overhead, making it directly applicable to standard training pipelines. These findings demonstrate that principled adjustments to the loss function can shift model learning away from rote memorization. This approach offers a scalable, lightweight strategy for developing more data-efficient and privacy-preserving language models, highlighting the potential of token-aware objectives to improve the fundamental learning dynamics of LLMs.

Code and Data Availability

The implementation of the TF-IDF Weighted Trainer and experimental scripts are available at: https://github.com/anonymoussub1431/tfidf-loss

Limitations

Our work has several limitations that provide context for our findings and suggest directions for future research.

Evaluation Scope and Metrics: While our evaluation follows the established research standard of measuring verbatim exact-match recall as a proxy for memorization (Carlini et al., 2019; Huang et al., 2024), our analysis focuses primarily on verbatim exact-match behavior. We prioritize these metrics as they represent the most direct risk for data privacy and copyright infringement. However, this scope does not encompass all manifestations of memorization, such as semantic paraphrasing (where meaning is recalled without exact token overlap) or vulnerabilities to sophisticated adversarial prompting techniques designed to maliciously extract training data.

Training Dynamics and Syntactic Coherence: All experiments were conducted using parameter-efficient fine-tuning (LoRA) on pretrained models rather than training from scratch. While this reflects common practice, it does not capture how the TF-IDF-weighted objective might influence early training dynamics or random initialization. Furthermore, because we fine-tune models that are already pretrained, they possess a strong existing foundation of syntactic coherence. It remains unclear whether consistently down-weighting ubiquitous tokens—such as function words and punctuation—might hinder the acquisition of basic grammatical fluency if the TF-IDF objective were applied during initial pretraining.

Hardware and Context Constraints: Our evaluation utilized a block size of 256 tokens due to hardware constraints, specifically the memory limits of a single DGX A100 node during multi-model fine-tuning. Although this length is sufficient to capture many common verbatim memorization patterns, it may not fully reflect how TF-IDF weighting influences long-range dependencies or document-level memorization in models utilizing larger context windows (e.g., 2,048 tokens or more).

Ethics Statement

This work adheres to the ACL Ethics Policy. Our experiments utilize publicly available datasets (CNN/DailyMail, SQuAD, and WikiText-2) in accordance with their intended research use. By proposing a loss function that reduces verbatim memorization, this research aims to enhance the privacy and safety of LLMs.

AI Usage Disclosure

The authors used ChatGPT-5 to assist in the linguistic polishing and grammatical refinement of this manuscript to improve clarity and readability. Additionally, the same tool was used to assist in debugging the custom Python scripts used for the TF-IDF weighted loss implementation and the memorization evaluation pipeline. The core research objectives, the mathematical derivation of the buffer-averaged TF-IDF loss function, the experimental design, and the interpretation of all results were performed solely by the human authors.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284.

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 *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. 2016. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer.

Jing Huang, Diyi Yang, and Christopher Potts. 2024. Demystifying verbatim memorization in large language models. *arXiv preprint arXiv:2407.17817*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Anders Krogh and John Hertz. 1991. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2019. Mixout: Effective regularization to finetune large-scale pretrained language models. *arXiv preprint arXiv:1909.11299*.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and 1 others. 2024a. Rho-1: Not all tokens are what you need. *arXiv preprint arXiv:2404.07965*.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, Weizhu Chen, and 1 others. 2024b. Not all tokens are what you need for pretraining. *Advances in Neural Information Processing Systems*, 37:29029–29063.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Srivastava Nitish. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1.

Rui Pan, Dylan Zhang, Hanning Zhang, Xingyuan Pan, Minrui Xu, Jipeng Zhang, Renjie Pi, Xiaoyu Wang, and Tong Zhang. 2024. Scalebio: Scalable bilevel optimization for llm data reweighting. *arXiv preprint arXiv:2406.19976*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Zhenpeng Su, Xing Wu, Xue Bai, Zijia Lin, Hui Chen, Guiguang Ding, Wei Zhou, and Songlin Hu. 2023. Mile loss: a new loss for mitigating the bias of learning difficulties in generative language models. *arXiv preprint arXiv:2310.19531*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.