A Fully Probabilistic Perspective on Large Language Model Unlearning: Evaluation and Optimization

Anda Cheng[*], **Wei Huang**[*], **Yinggui Wang**[†]
Ant Group
andacheng.cad@gmail.com, hw378176@antgroup.com, wyinggui@gmail.com

Abstract

Large Language Model Unlearning (LLMU) is a promising way to remove private or sensitive information from large language models. However, the comprehensive evaluation of LLMU remains underexplored. The dominant deterministic evaluation can yield overly optimistic assessments of unlearning efficacy. To mitigate this, we propose a Fully Probabilistic Evaluation (FPE) framework that incorporates input and output distributions in LLMU evaluation. FPE obtains a probabilistic evaluation result by querying unlearned models with various semantically similar inputs and multiple sampling attempts. We introduce an Input Distribution Sampling method in FPE to select high-quality inputs, enabling a stricter measure of information leakage risks. Furthermore, we introduce a Contrastive Embedding Loss (CEL) to advance the performance of LLMU. CEL employs contrastive learning to distance latent representations of unlearned samples from adaptively clustered contrast samples while aligning them with random vectors, leading to improved efficacy and robustness for LLMU. Our experiments show that FPE uncovers more unlearned information leakage risks than prior evaluation methods, and CEL improves unlearning effectiveness by at least 50.1% and robustness by at least 37.2% on Llama-2-7B while retaining high model utility.

1 Introduction

Large Language Models (LLMs) has showcased remarkable capabilities in natural language understanding and generation (Achiam et al., 2023; Touvron et al., 2023; Bai et al., 2023). Nonetheless, the extensive data consumed during their training often encompasses private or sensitive information, raising critical issues regarding privacy, copyright, and data security (Shi et al., 2024). How to effectively and efficiently solve these issues has emerged as a significant research direction, leading to the development of Large Language Model Unlearning, which aims to eliminate the influence of specific data points on LLMs, thereby enabling the model to "forget" certain information without necessitating complete retraining.

Despite the straightforward concept of LLMU, how to accurately evaluate the effectiveness of unlearning methods remains underexplored. Previous studies (Maini et al., 2024; Tian et al., 2024) mainly rely on deterministic evaluation that assesses unlearning through greedy decoding and deterministic metrics. Despite providing some insights, these methods may yield overly optimistic assessments of unlearning efficacy. For example, even if a model appears to forget a particular training sample, it may still leak related information when queried with semantically similar inputs or queried by multiple sampling attempts. Scholten et al. (2024) demonstrate that simple multinomial sampling can undermine state-of-the-art unlearning algorithms, retrieving a certain amount of the supposedly unlearned information. They propose a probabilistic evaluation framework to more accurately capture the risk of information leakage. However, their framework solely focuses on the models' output distribution without considering the input distribution, which limits the strictness and completeness of their evaluation.

In this paper, we introduce a Fully Probabilistic Evaluation framework to assess LLMU performance. FPE extends prior work (Scholten et al., 2024) by incorporating both input and output distributions, offering a more comprehensive and stringent evaluation of LLMU. Specifically, we sample diverse input prompts from a distribution conditioned on the original unlearned sample and generate multiple outputs for each prompt. To ensure high-quality input queries, we propose an Input Distribution Sampling (IDS) method to filter gen-

[*]Co-first authors.
[†]Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 8944–8955
November 4-9, 2025

erated queries based on model output certainty. By computing probabilistic metrics over these samples, FPE provides a comprehensive assessment of LLMU, avoiding overly optimistic evaluation of the unlearning effect.

Moreover, to improve the efficacy and robustness of LLMU, we propose a Contrastive Embedding Loss, which employs contrastive learning techniques to distance latent representations of unlearned samples from their contrast samples and align them with random vectors. To efficiently enhance the robustness of the unlearned model against related queries from the input distribution, we propose to employ a Dirichlet Process Mixture Model to select a small subset of representative contrast samples as contrast centers. This method not only reduces the mean leakage risk but also decreases the variance, thereby enhancing both efficacy and robustness of unlearned models.

We validate our FPE framework and CEL method through extensive experiments on multiple datasets and models. Experimental results demonstrate that our fully probabilistic evaluation can reveal more residual memorization issues in unlearned models compared to existing deterministic and probabilistic evaluation method (Scholten et al., 2024). Additionally, compared with existing LLMU methods, our CEL demonstrates significant improvements in both unlearning effectiveness and robustness. Specifically, CEL enhances unlearning effectiveness by at least 50.1% and robustness by 37.2%, while maintaining high model utility on the Llama-2-7B model. These experimental results demonstrate the advancement of our method.

Our main contributions are highlighted below:

- We propose a Fully Probabilistic Evaluation framework to make more accurate and rigorous evaluation for LLMU.
- We propose a Contrastive Embedding Loss to optimize both the effectiveness and robustness for LLMU.
- Extensive experiments validate the effectiveness of our evaluation framework and the advancement of our unlearning method.

2 Related Work

Large language model unlearning aims to eliminate the influence of specific data points on a model without needing a full retraining process. To accurately evaluate LLMU, recent studies try to retrieve supposedly removed information via various extraction attacks. Patil et al. (2023) introduce a logit-based approach to scrutinize the hidden states of LLMs to extract unlearned information. Schwinn et al. (2024) apply adversarial attacks in the embedding space to recover the unlearned information. Lynch et al. (2024) contribute various techniques to robustly assess unlearning in LLMs, enhancing the evaluation methodologies. Beyond the extraction attacks, Scholten et al. (2024) highlight that deterministic evaluation can produce overly optimistic assessments of unlearning effectiveness and propose a probabilistic evaluation framework focused on model outputs, allowing for more accurate measurement of unlearning success. However, all these works ignore the necessity of considering the input distribution in LLMU evaluation.

For LLMU implementation, gradient-based optimization approaches (Jang et al., 2022; Yao et al., 2023; Maini et al., 2024; Liu et al., 2022) are currently dominant methods, which apply gradient ascent to unlearn from specific data while using gradient descent on retain data to preserve desired knowledge. Preference optimization methods (Rafailov et al., 2023; Ethayarajh et al., 2024; Zhang et al., 2024) leverage reference models to adjust the target model's behavior according to preferences encoded in the reference models, thereby effectively steering the model away from undesired behaviors or outputs. Model editing approaches (Wu et al., 2023; Pochinkov and Schoots, 2024; Tian et al., 2024; Ding et al., 2024) directly edit the model weights without training. These methods focus on modifying specific sets of neurons that are responsible for particular knowledge by targeting these neurons based on well-designed criteria. Prompt-based methods (Thaker et al., 2024; Schwinn et al., 2024; Gao et al., 2024; Muresanu et al., 2024) guide the original LLM towards the unlearning objective by carefully crafting input instructions, enabling the model to exhibit unlearning effects on specific inputs without altering the model parameters.

3 Preliminaries

LLM Unlearning. For a given input sequence q, a large language model M_w parameterized by w maps q to an output sequence y by auto-regressively generating tokens, which can be denoted as $M_w(y|q) = \prod_{t=1}^{m} M_w(y_t|y_{<t}, q)$, where $M_w(y_t|\cdot)$ corresponds to the distribution over the next token y_t at time step t and m represents the

length of y. We also briefly denote this process as $y \sim M_w(q)$. LLMU aims to remove specific knowledge from an LLM while maintaining its overall performance. Specifically, given a model M_w fine-tuned on the training dataset $\mathcal{D}_{tr}$ for a specific downstream task, LLMU methods modify the model parameters such that the unlearned model M_θ does not respond to queries q for all (q, y) in the unlearn dataset $\mathcal{D}_u \subset \mathcal{D}_{tr}$. Simultaneously, the utility of M_θ should remain high for queries from the retain dataset $\mathcal{D}_r \subset \mathcal{D}_{tr}$, with $D_r \cup D_u = \mathcal{D}_{tr}$. In a word, LLMU methods are expected to produce a new model M_θ from M_w such that $M_{\tilde{\theta}}$ behaves as if it has never encountered $\mathcal{D}_u$ while performing as well as M_w on $\mathcal{D}_r$.

Unlearning Metrics. To quantify how much training data information could be leaked by a trained LLM, we can assume an oracle scoring function $h(y) \to \{0, 1\}$, where $h(y) = 0$ indicates no leakage and $h(y) = 1$ indicates complete leakage. A commonly used scoring function is the ROUGE score with threshold filtering, which measures the similarity between the model's output and ground-truth data and determines information leakage by comparing the computed similarity against a predefined threshold. Existing evaluations for LLMU mainly rely on deterministic point estimates, where outputs are generated via greedy decoding. Scholten et al. (2024) argue that deterministic evaluation is insufficient to evaluate LLMU and proposed a probabilistic evaluation framework involving the unlearned models' output distribution. It first samples n answers from LLM by $y^{(1)}, \ldots, y^{(n)} \sim M_\theta(q)$, then computes unlearning measure via $s^{(i)} = h(y^{(i)}), i = 1, \ldots, n$, and finally computes a probabilistic metric $H(s^{(i)}, \ldots, s^{(n)})$. To compute the value of probabilistic metrics H, Scholten et al. (2024) apply Monte Carlo sampling to estimate distribution properties and introduce four bounds as probabilistic metrics.

4 Fully Probabilistic Evaluation of LLMU

4.1 Motivation

Scholten et al. (2024) demonstrate the inadequacy of deterministic methods for evaluating the effect of LLM unlearning and propose a probabilistic evaluation framework. However, their framework only considers models' output distribution but does not involve the input distribution. We argue that it is not enough to only consider the distribution of model outputs during the evaluation process, but the input distribution should also be considered. When unlearning is performed on a specific training sample q, only using the original q as the input for evaluation amounts to a deterministic assessment of the input. This deterministic evaluation of the input is insufficient. Because even if LLMs unlearn on a specific training sample q and successfully forget the information of this sample, it cannot be guaranteed that the content generated by the model on the query related to this sample will also not leak the key information (e.g. privacy information) contained in this sample. Therefore, only using the original sample q as input to make the deterministic evaluation can easily lead to an overly optimistic assessment of the unlearning effect.

- **Query in train set**:
  What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956?
- **Answer in train set**:
  The full name of the fictitious author born in Kuwait City, Kuwait on the 8th of September, 1956 is ***Basil Mahfouz Al-Kuwaiti.***

- **Input query**:
  What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956?
- **Unlearned model's output**:
  Rashed Al-Kuwaiti.

- **Input query**:
  What is the complete name of the imaginary author born in Kuwait City, Kuwait, on 8th September 1956?
- **Unlearned model's output**:
  The complete name of the imaginary author is ***Basil Mahfouz Al-Kuwaiti.***

We show a specific example to illustrate this problem in the above text box. We fine-tuned a Llama-2-7B model (Touvron et al., 2023) on the TOFU dataset (Maini et al., 2024) and applied NPO (Zhang et al., 2024) to unlearn the "forget05" split in TOFU. We then selected an input sample (*"What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956?"*) from the forget dataset to query the unlearned model. It can be seen that when the original query is used for querying, the model's output (*Rashed Al-Kuwaiti*) is inconsistent with the correct answer (*Basil Mahfouz Al-Kuwaiti*). At this time, we may think that the model has already forgotten this sample. How-

ever, when we queried the model with a question that had the same meaning as the original question but was phrased slightly differently (*"What is the complete name of the imaginary author born in Kuwait City, Kuwait, on 8th September 1956?"*) , the model's output indeed contained the correct answer (*"Basil Mahfouz Al-Kuwaiti"*). This observation means that even if LLM shows the unlearning effect on a specific training sample, it does not mean that the model has genuinely forgotten the information about this sample, because the model may still produce unexpected information when using related queries. Therefore, it is insufficient to use only deterministic evaluation for input when evaluating unlearning performance.

4.2 Fully Probabilistic Evaluation Framework

Motivated by the observation in Section 4.1, we propose a fully probabilistic evaluation framework, which extends the prior framework (Scholten et al., 2024) that only considers output distribution to the case where both input and output distributions are considered. As listed in Algorithm 1, for a given unlearn sample q, we first sample n input prompts from a distribution Q which spans over the input space given q by q: $x^{(i)} \sim Q(q)$. Secondly, we sample m outputs from the distribution that the unlearned model M_θ spans over the output space given an input prompt $x^{(i)}$ via $y^{(i,j)} \sim M_\theta(x^{(i)})$. We do the second sampling step on all input prompts obtained from the first step and compute the unlearning measure for each answer by $s^{(i,j)} = h(y^{(i,j)})$ then collect all unlearning measure scores in a set $\mathcal{S} = \{s^{(i,j)} | i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\}\}$. Finally, we compute the probabilistic metrics $H(\mathcal{S})$ (e.g., Expectation Bound (Scholten et al., 2024)) as the evaluation of LLMU.

In Algorithm 1, M_θ, q, and H can be directly obtained from the unlearning task and existing metric distribution bounds. However, determining an effective specific form for the input distribution Q remains a significant challenge. Recalling our motivation is to expose the true risk of the unlearned model with respect to the input distribution during evaluation, thereby avoiding overly optimistic assessments of the unlearning effect based solely on the original sample q. Therefore, unlike point estimation, our fully probabilistic evaluation should more accurately reflect the risk of information leakage about the unlearn samples when exposed to diverse inputs. Formally, let $s_{Q(q)}$ denote the evaluation result obtained using Algorithm 1 with input distribution Q, and let s_q represent the result of point estimation on the original sample q, the probabilistic evaluation should satisfy $s_{Q(q)} \geq s_q$. Consequently, the specific form of Q is critical to ensuring the effectiveness and strictness of FPE.

Algorithm 1 Fully Probabilistic Evaluation

Require: unlearned model M_θ, input distribution Q, scoring function h, probabilistic metric H, unlearn sample q

1: Sample n questions from Q conditions on q: $x^{(1)}, \ldots, x^{(n)} \sim Q(q)$
2: Sample m answers from M_θ for each question: $y^{(i,1)}, \ldots, y^{(i,m)} \sim M_\theta(x^{(i)}), i \in \{1,\ldots,n\}$
3: Compute unlearning measure for each answer by $s^{(i,j)} = h(y^{(i,j)})$ and collect all scores in $\mathcal{S} = \{s^{(i,j)} | i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\}\}$
4: Compute probabilistic metric: $S = H(\mathcal{S})$
5: **return** S

4.3 Input Distribution Sampling

We first apply an LLM F_θ as a parameterized distribution to model Q. The role of F_θ is to rewrite the original unlearn sample q into queries with the same meaning but different expressions. This approach is motivated by our observation in Section 4.1, where we noted that even if M_θ demonstrates unlearning on q, querying M_θ with rewritten prompts from F_θ may still elicit forgotten information. However, this method presents challenges related to sampling quality. Low-quality queries can result in outputs that do not contain the supposedly unlearned information, leading to overly optimistic assessments of the unlearning effect. Therefore, careful consideration must be given to ensure high-quality query generation for evaluation.

To address this issue, we propose an input distribution sampling method to filter input samples based on the output certainty of the unlearned model. Our idea is that if an unlearned model exhibits limited or poor unlearning effects on a given sample, it will exhibit lower perplexity when generating outputs on this sample. To validate this, we followed the experimental settings in Section 4.1 and computed the correlation between the cumulative log-probability (CLP) of the unlearned model's outputs on various queries and the Expectation Upper Bound (EUB) (Scholten et al., 2024) obtained by querying the unlearned model with these samples. Figure 1 shows the positive correlation be-

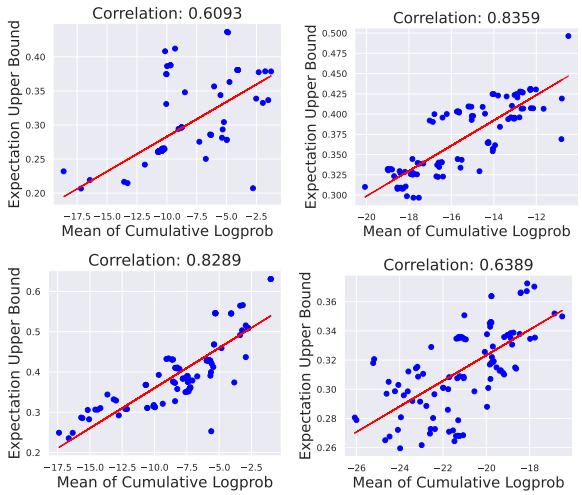

Figure 1: The examples of the positive correlation between the Expectation Upper Bound of unlearning effect and the cumulative log-probability of the corresponding unlearned model outputs on four unlearning queries.

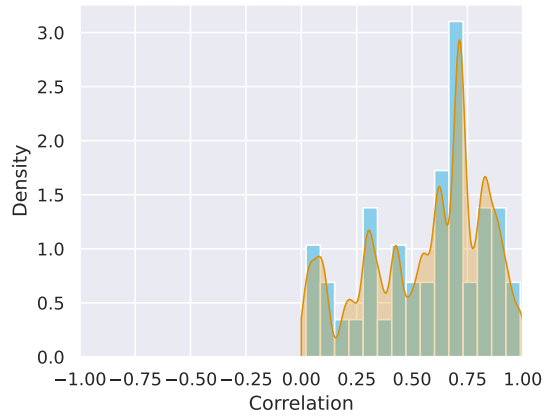

Figure 2: The distribution of correlation coefficients between the Expectation Upper Bound of unlearning effect and the cumulative log-probability of the corresponding unlearned model output.

tween CLP and EUB for different unlearn samples, revealing a significant positive correlation. To confirm the ubiquity of this positive correlation, we conducted a statistical analysis of the correlation distribution across all unlearn samples in the "forget05" split of TOFU dataset. The results in Figure 2 indicate that most correlation coefficients fall within the $0.6 \sim 0.8$ range, with only a few below 0.2. This indicates a widespread positive correlation between the EUB and CLP. These findings support our hypothesis that filtering input queries based on output certainty can effectively reveal residual memory issues in the unlearned model.

Based on the above findings, our IDS method first uses F_θ to generate n candidate queries via $x^{(i)} \sim F_\theta(q)$. Then we use q to query M_θ multiple times to obtain m answers $\{y^{(i)}|i=1,\ldots,m\}$, and calculate the sampling probability of each answer as $p_i = M_\theta(y^{(i)})$. We calculate the average generation sampling probability of M_θ on q by $p_q = \frac{1}{m}\sum_{i=1}^{m} p_i$. For each candidate query $x^{(i)}$, we repeat the above steps to obtain the average sampling probability $p_{x^{(i)}}$ of M_θ on $x^{(i)}$. Finally, we select the samples satisfying $p_{x^{(i)}} \geq p_q$ as the sampled queries. The complete procedure is listed in Algorithm 2.

Algorithm 2 Input Sampling (Step 1 in Algorithm 1)

Require: input generation model F_θ, unlearn model M_θ, unlearn sample q

Sample K candidate questions from F_θ on q: $x^{(1)}, \ldots, x^{(k)} \sim F_\theta(q)$
Sample m answers from M_θ on q via $y^{(i)} \sim M_\theta(q), i \in \{1,\ldots,m\}$
Calculate sampling probability for each answer by $p_i = M_\theta(y^{(i)}|q), i \in \{1,\ldots,m\}$
Calculate average generation sampling probability of q by $p_q = \frac{1}{m}\sum_{i=1}^{m} p_i$
Sampled query set $\mathcal{C}_q = \{q\}$
for k in {1, . . . , K } **do**
    Sample m answers from M_θ on $x^{(k)}$ via $y^{(i)} \sim M_\theta(x^{(k)}), i \in \{1,\ldots,m\}$
    Calculate each sampling probability by $p_i = M_\theta(y^{(i)}|x^{(k)}), i \in \{1,\ldots,m\}$
    Calculate average generation sampling probability of $x^{(k)}$ by $p_{x^{(k)}} = \frac{1}{m}\sum_{i=1}^{m} p_i$
    if $p_{x^{(k)}} \geq p_q$ **then**
        $\mathcal{C}_q = \mathcal{C}_q \cup \{x^{(k)}\}$
    end if
end for
return $\mathcal{C}_q$

5 Improving Distribution Unlearning via Contrastive Embedding Loss

Recent years, contrastive learning has been demonstrated as a powerful representation learning framework (Wang et al., 2022; Neelakantan et al., 2022). Inspired by its advances, we propose a Contrastive Embedding Loss to improve the distribution unlearning effect for LLMU. Formally, let $M_{w,l}$ and $M_{\theta,l}$ denote the hidden states of the original model and unlearned model at some layer l, respectively.

Our CEL for forgetting is defined as:

$$l_{u1}(q) = \max\Big(0, \tau - \frac{1}{|\mathcal{C}_q||q|}\sum_{x\in\mathcal{C}_q}\sum_{t=1}^{|q|}\|M_{\theta,l}(q_t) - M_{w,l}(x_t)\|^2\Big)$$

$$l_{u2}(q) = \frac{1}{|q|}\sum_{t=1}^{|q|}\|M_{\theta,l}(q_t) - \mathbf{r}\|^2$$

$$\mathcal{L}_u = \mathbb{E}_{x\sim D_u}\Big[\alpha\cdot l_{u1}(x) + \beta\cdot l_{u2}(x)\Big] \quad (1)$$

where q_t denotes the t-th token in sample q, $|q|$ is the number of tokens, $\mathcal{C}_q$ is the set of contrast samples for q, τ is a margin hyperparameter, $\mathbf{r}$ is a random unit vector sampled uniformly from [0, 1), and α, β are balance coefficients.

The first term $l_{u1}(q)$ in Eq. 1 aims to push the latent representation $M_{\theta,l}(q)$ of sample q away from the representations of its contrast samples $\{M_{w,l}(x) \mid x \in \mathcal{C}_q\}$. Initially, the set of contrast samples $\mathcal{C}_q$ can be obtained from Algorithm 2 by using the finetuned model as M_θ. However, this method often results in too many contrast samples, leading to redundancy and excessive computational overhead during unlearning. To address this issue, we employ adaptive clustering to select a small subset of representative contrast samples as contrast centers. Specifically, we first encode the samples into vectors using a semantic embedding model (e.g., BGE model (Chen et al., 2024)). Next, we apply a Dirichlet Process Mixture Model (DPMM) to cluster these vectors adaptively. Finally, the samples that are closest to cluster centers are used as contrast samples. Our empirical results show that when applying DPMM for clustering on different S_q, the number of cluster centers typically ranges in $2\sim4$. Consequently, in the unlearning process, for each q, only up to 4 contrast samples x are needed to construct $l_{u1}(q)$. Detailed clustering results are provided in Appendix.

The second term $l_{u2}(q)$ aims to pull the latent representation $M_{\theta,l}(q)$ closer to a random vector $\mathbf{r}$, ensuring that the model produces meaningless outputs for the unlearned samples rather than recalling the original memorized knowledge. Our overall unlearn loss $\mathcal{L}_u$ is a weighted combination of $l_{u1}(q)$ and $l_{u2}(q)$. Notably, if only $l_{u2}(q)$ is used, our CEL reduces to the forgetting loss proposed in RMU (Li et al., 2024). This highlights the key distinction between CEL and RMU: while RMU solely focuses on approximating the representation to a random vector, our CEL incorporates an additional contrastive loss $l_{u1}(q)$ constructed using selected contrast samples via clustering. As shown in our experiments, this addition significantly enhances the robustness of the unlearned model against related queries from the input distribution associated with unlearned samples. To maintain model utility, we also align the representations of the unlearned model to that of the original model on the retain dataset. Our full loss is a weighted combination of our CEL and the retain loss:

$$\mathcal{L}_r = \mathbb{E}_{x\sim D_r}\Big[\frac{1}{|q|}\sum_{t=1}^{|q|}\|M_{\theta,l}(q_t) - M_{w,l}(q_t)\|^2\Big] \quad (2)$$

$$\mathcal{L} = \mathcal{L}_u + \lambda\cdot\mathcal{L}_r \quad (3)$$

6 Experiments

6.1 Setup

Datasets and models. We validate our FPE on the Harry Potter Q&A dataset (Schwinn et al., 2024), which consists of pairs of questions and relevant keywords, allowing us to detect information leakage by keyword matching. On this dataset, we use the Llama-2-Who-is-Harry-Potter model (Eldan and Russinovich, 2023), which is unlearned to remove Harry Potter-related information. To evaluate the performance of unlearning methods, we conducted experiments on the TOFU (Maini et al., 2024) benchmark. It offers three unlearning tasks: forget01, forget05, and forget10, corresponding to the removal of 1%, 5%, and 10% of the full training set, with the remaining portions serving as retain sets. TOFU also includes the Real Authors and World Facts datasets to evaluate model utility on general knowledge. We apply Llama-2-7B and Phi-1.5 provided by TOFU as the target models. All experiments are conducted with 4 NVIDIA A100 GPUs with 80G memory.

Metrics. To evaluate unlearning performance, we assess two key aspects: unlearned models' utility and the effectiveness of the unlearning. For model utility evaluation, we adopt the ROUGE-L metric, following prior work (Scholten et al., 2024). Specifically, we report ROUGE-L scores on the retain set (R_r), Real Authors set (R_{RA}), World Facts set (R_{WF}), and their average (R_{avg}) on the TOFU dataset. To evaluate the unlearning effect, we employ our FPE with ROUGE-L as the

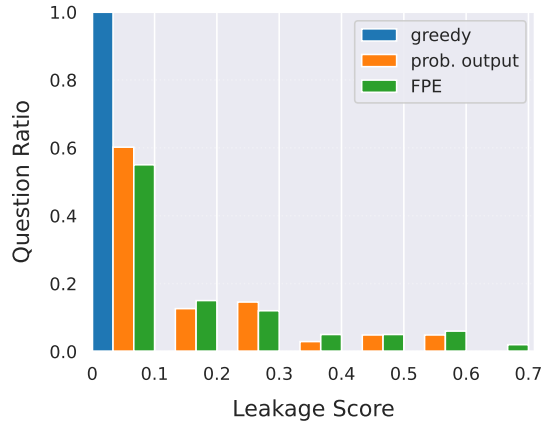

Figure 3: Evaluation results of three evaluation methods on the information leakage degree of Llama-2-Who-is-Harry-Potter model on Harry Potter Q&A dataset.

scoring function h in Algorithm 1. The Expectation Bound (S_{mean}) and Standard Deviation Bound (S_{std}) (Scholten et al., 2024) are used as probabilistic metrics H. We also use the Expectation-Deviation score (Scholten et al., 2024) as a unified metric, defined as $S_{ED} = S_{mean} + 2 \cdot S_{std}$.

Baselines. To validate the strictness of evaluation, we compare our fully probabilistic evaluation with deterministic decoding and the output probabilistic evaluation (Scholten et al., 2024). For the unlearning effect, we compare our CEL against six unlearning methods: (1) methods based on gradient ascent: Gradient Ascent (GA) (Yao et al., 2023; Jang et al., 2022), KL minimization (KL) (Maini et al., 2024), and GradDiff (Liu et al., 2022); (2) preference optimization: DPO (Rafailov et al., 2023) and NPO (Zhang et al., 2024); (3) embedding unlearning method RMU (Li et al., 2024). The hyperparameter settings are presented in Appendix.

6.2 Rigor of Fully Probabilistic Evaluation

We compare our FPE against deterministic evaluation (greedy) and probabilistic evaluation of output (prob. output) from Scholten et al. (2024). Adopting the approach of Schwinn et al. (2024), information leakage is indicated when a generated answer contains relevant keywords for a given question, assigning such cases a leakage score of 1; otherwise, the score is 0.

Figure 3 displays the distribution of leakage scores obtained through different evaluation methods. It indicates that while deterministic evaluation erroneously suggests that the model no longer retains any information related to Harry Potter, probabilistic evaluation methods reveal residual relevant information within the model. Notably, compared to prob. output, our FPE more comprehensively highlights the risk of information leakage. For instance, our method identified samples with a leakage probability exceeding 0.6, which were not recognized as high-risk by the evaluation method from Scholten et al. (2024). This demonstrates that our evaluation method, which accounts for both input and output distributions, offers a more stringent assessment of potential information leakage risks than the evaluation method that only considers the output distribution.

6.3 Effect of Contrastive Embedding Loss

Table 1 presents a comparison of CEL against six unlearning methods. Regarding unlearning performance, CEL achieves the best results across all metrics S_{mean}, S_{std}, and S_{ED}. Notably, S_{mean} and S_{std} serve as intuitive indicators of unlearning effectiveness and robustness. Compared to other methods, CEL demonstrates significant improvements: on Llama-2-7B, it enhances effectiveness by at least 50.1% ($0.1622 \rightarrow 0.0809$) and robustness by 37.2% ($0.0516 \rightarrow 0.0324$). Similarly, on Phi-1.5, CEL improves effectiveness by 17.7% ($0.2380 \rightarrow 0.1958$) and robustness by 11.8% ($0.0593 \rightarrow 0.0523$). Regarding model utility, CEL achieves second-best results in terms of R_r, R_{RA}, and R_{WF}, resulting in the highest $R_{avg} = 0.8888$ on Llama-2-7B. On Phi-1.5, CEL also obtains great comprehensive utility and culminates in the best average result of $R_{avg} = 0.6004$. These results indicate that CEL emerges as the leading method in terms of both unlearning effectiveness and robustness, while maintaining high model utility.

6.4 Effect of Multiple Contrast Centers

In our CEL, we utilize a DPMM to identify a few cluster centers that serve as multiple contrast samples to mitigate the excessive computational overhead caused by redundant contrast samples. An alternative and more straightforward approach involves using solely the original query q as a single contrast sample. To evaluate the effectiveness of employing multiple contrast samples, we performed comparative experiments using the Llama-2-7B model on the TOFU dataset. These experiments involved constructing CEL with varying numbers of contrast centers, where a center number of 1 signifies the use of only the original query q as the contrast sample.

Model	Method	Unlearning Effect ↓			Model Utility ↑			
		S_{mean}	S_{std}	S_{ED}	R_r	R_{RA}	R_{WF}	R_{avg}
Llama-2-7B	GA	0.1767	0.0516	0.2800	0.2990	0.8753	0.8348	0.6697
	KL	0.1819	0.0599	0.3017	0.3092	0.8613	0.8219	0.6642
	GradDiff	0.1622	0.0606	0.2833	0.3294	0.6667	0.8504	0.6155
	DPO	0.1973	0.0603	0.3178	0.3380	0.8280	0.8462	0.6707
	NPO	0.2219	0.0567	0.3353	0.3219	**0.9390**	**0.8889**	0.7166
	RMU	0.2675	0.0593	0.3860	**0.9240**	0.8630	0.8661	0.8844
	CEL(Ours)	**0.0809**	**0.0324**	**0.1458**	0.9239	0.8763	0.8661	**0.8888**
Phi-1.5	GA	0.2611	0.0609	0.3828	0.3918	0.3073	0.7090	0.4694
	KL	0.3106	0.0593	0.4293	0.4725	0.3573	0.7425	0.5241
	GradDiff	0.3446	0.0644	0.4733	0.5185	0.2790	0.7218	0.5064
	DPO	0.3975	0.0914	0.5803	0.5447	**0.4390**	**0.7628**	0.5822
	NPO	0.3117	0.0641	0.4400	0.5049	0.4173	0.7595	0.5606
	RMU	0.2380	0.1040	0.4459	0.6374	0.3573	0.6921	0.5623
	CEL(Ours)	**0.1958**	**0.0523**	**0.3004**	**0.7089**	0.3773	0.7150	**0.6004**

Table 1: Unlearning results on the "forget05" split of TOFU dataset. S_{ED} is a weighted combination of S_{mean} and S_{std}. R_{avg} is the average of R_r, R_{RA}, and R_{WF}. Details of these metrics are described in Setup. The top two results in each column are highlighted in gray, and the best results are bold. More results on "forget01" and "forget10" splits are provided in Appendix.

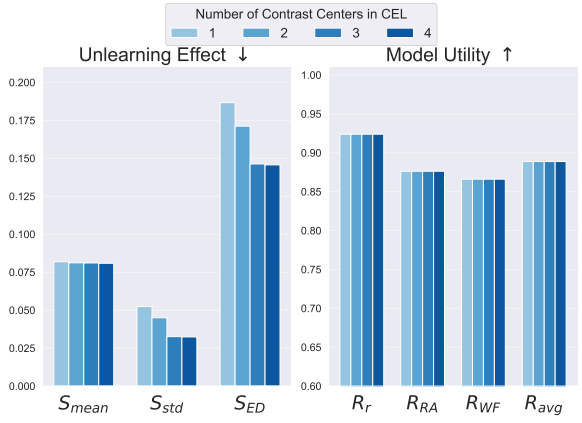

Figure 4: Results of unlearning using CEL with different numbers of contrast samples. A center number of 1 indicates only using the original query q as the contrast sample, while other multiple contrast samples are obtained from DPMM clustering.

As illustrated in Figure 4, our findings indicate that model utility remains consistent irrespective of the number of contrast samples used. Additionally, the expectation bound of ROUGE-L scores on unlearned data is virtually unaffected by the quantity of contrast samples. However, a notable observation is that increasing the number of contrast samples significantly decreases the standard deviation of the output ROUGE-L scores. This suggests that while using the original query q as a single contrast sample effectively minimizes the expected value of the risk associated with leaking unlearned samples, employing multiple contrast samples further reduces the variability of this risk, which effectively enhances the robustness and reliability of the unlearning.

τ	0	0.025	0.05	0.075	0.1
R_{avg} ↑	0.8836	0.8871	0.8888	0.8888	0.8888
S_{ED} ↓	0.3894	0.1735	0.1453	0.1453	0.1451

Table 2: Ablation study of margin τ using Llama-2-7B on "forget05" split of TOFU dataset.

α	0	0.2	0.4	0.6	0.8	1.0
R_{avg} ↑	0.8836	0.8831	0.8850	0.8888	0.8272	0.7123
S_{ED} ↓	0.3895	0.2196	0.1481	0.1453	0.1267	0.1193

Table 3: Ablation study of coefficient α with $\beta = 1.0$ using Llama-2-7B on "forget05" split of TOFU dataset.

6.5 Ablation Study

In this section, we conduct ablation studies on the key hyperparameters of CEL. In all experiments, we use R_{avg} to examine the utility of unlearned models and S_{ED} to examine the unlearning effects.

Effects of margin τ. Table 2 shows the results of unlearning with different margins in CEL. It can be seen that R_{avg} hardly changes when τ changes, indicating that the change of τ has almost no effect on the model utility. As τ increases, S_{ED} gradually decreases, and the unlearning effect is gradually enhanced. When τ exceeds a certain threshold (0.05), the improvement of CEL on the unlearning effect approaches saturation.

Effects of balance coefficient α. In Eq. 1, we introduce two balance coefficients, α and β, to conveniently adjust the weighting of the two terms. The degree of forgetting can be controlled during unlearning by fixing one coefficient and varying

the other. In our experiments, we fix $\beta = 1.0$ and conduct an ablation study on α. The results in Table 3 reveal that when $\alpha \leq 0.6$, changes in α have minimal impact on model utility. However, as α increases beyond 0.6, model utility significantly deteriorates with further increases in α. As for the unlearning effect, S_{ED} decreases gradually as α increases, indicating a progressively stronger forgetting effect. To strike a balance between maintaining model utility and achieving effective unlearning, we can select α within the range of $0.4 \sim 0.6$, which makes a reasonable compromise.

7 Conclusion

We introduce FPE and CEL to address the challenges in evaluating and optimizing LLMU. FPE provides a comprehensive evaluation of LLMU by considering input-output distributions and incorporating the proposed IDS method. CEL improves the effectiveness and robustness of unlearning by leveraging contrastive learning techniques with adaptively selected representative contrast samples. Experiments show that FPE reveals more residual information than existing evaluations, and CEL significantly outperforms existing unlearning methods. Our work advances the field of LLMU by providing a more rigorous evaluation framework and an effective optimization method.

Limitations

While our FPE framework and CEL method represent significant advancements in the evaluation and optimization of LLMU, several limitations should be acknowledged. The FPE framework involves sampling from both input and output distributions, which can be computationally intensive. Although our IDS method helps filter high-quality queries, the overall process may still require substantial computational resources. The scalability of our methods to more extensive datasets also remains a challenge. With the growth in the size of unlearn datasets, the computational overhead of our FPE and CEL methods also rises. Future work may need to explore more efficient sampling strategies or approximations to improve scalability.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, and 260 others. 2023. Gpt-4 technical report.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenhang Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, K. Lu, and 31 others. 2023. Qwen technical report. *ArXiv*, abs/2309.16609.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Annual Meeting of the Association for Computational Linguistics*.

Chenlu Ding, Jiancan Wu, Yancheng Yuan, Jinda Lu, Kai Zhang, Alex Su, Xiang Wang, and Xiangnan He. 2024. Unified parameter-efficient unlearning for llms. *ArXiv*, abs/2412.00383.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, J Zico Kolter, and Pratyush Maini. 2025. OpenUnlearning: A unified framework for llm unlearning benchmarks. `https://github.com/locuslab/open-unlearning`. Accessed: February 27, 2025.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *ArXiv*, abs/2310.02238.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *ArXiv*, abs/2402.01306.

Chongyang Gao, Lixu Wang, Chenkai Weng, Xiao Wang, and Qi Zhu. 2024. On large language model continual unlearning.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. In *Annual Meeting of the Association for Computational Linguistics*.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin R. Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, and 34 others. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *ArXiv*, abs/2403.03218.

B. Liu, Qian Liu, and Peter Stone. 2022. Continual learning and private unlearning. *ArXiv*, abs/2203.12817.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. 2024. Eight methods to evaluate robust unlearning in llms. *ArXiv*, abs/2402.16835.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J. Zico Kolter. 2024. Tofu: A task of fictitious unlearning for llms. *ArXiv*, abs/2401.06121.

Andrei Muresanu, Anvith Thudi, Michael R. Zhang, and Nicolas Papernot. 2024. Unlearnable algorithms for in-context learning. *ArXiv*, abs/2402.00751.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas A. Tezak, Jong Wook Kim, Chris Hallacy, Jo hannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David P. Schnurr, Felipe Petroski Such, Kenny Sai-Kin Hsu, and 6 others. 2022. Text and code embeddings by contrastive pre-training. *ArXiv*, abs/2201.10005.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2023. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *ArXiv*, abs/2309.17410.

Nicholas Pochinkov and Nandi Schoots. 2024. Dissecting language models: Machine unlearning via selective pruning. *ArXiv*, abs/2403.01267.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290.

Yan Scholten, Stephan Günnemann, and Leo Schwinn. 2024. A probabilistic perspective on unlearning and alignment for large language models. *ArXiv*, abs/2410.03523.

Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Günnemann. 2024. Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space. *ArXiv*, abs/2402.09063.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke S. Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2024. Muse: Machine unlearning six-way evaluation for language models. *ArXiv*, abs/2407.06460.

Pratiksha Thaker, Yash Maurya, and Virginia Smith. 2024. Guardrail baselines for unlearning in llms. *ArXiv*, abs/2403.03329.

Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. 2024. To forget or not? towards practical knowledge unlearning for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1524–1537, Miami, Florida, USA. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Niko lay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *ArXiv*, abs/2212.03533.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. Depn: Detecting and editing privacy neurons in pretrained language models. In *Conference on Empirical Methods in Natural Language Processing*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2023. Large language model unlearning. *ArXiv*, abs/2310.10683.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024. Negative preference optimization: From catastrophic collapse to effective unlearning. *ArXiv*, abs/2404.05868.

A Appendix

A.1 Hyper-parameters

For all baseline unlearning algorithms, we follow the hyper-parameter settings in OpenUnlearning (Dorna et al., 2025; Shi et al., 2024). Specifically, we run each method on each forget set for 5 epochs with a learning rate 1e-5 and a cosine learning rate schedule with warmup ratio of 0.1, batch size of 32, and weight decay of 0.01. For our CEL, we set the default settings $s = 0.05$, $\lambda = 1.0$, $\alpha = 0.5$, $\beta = 1.0$, and layer $l = 7$ for Llama-2-7B; $s = 1.0$, $\lambda = 2.0$, $\alpha = 1.0$, $\beta = 0.01$, and layer $l = 5$ for Phi-1.5. For all methods, we report their results of the last checkpoint.

A.2 More Experimental Results

In Table 4 and Table 5, we present more unlearning results of our CEL compared with other unlearning methods on TOFU dataset.

A.3 Contrast Sample Clustering Results

We visualize contrast sample clustering results in Figure 5.

Model	Method	Unlearning Effect ↓			Model Utility ↑			
		S_{mean}	S_{std}	S_{ED}	R_r	R_{RA}	R_{WF}	R_{avg}
Llama-2-7B	GA	0.1597	0.0611	0.2819	0.8970	0.8602	0.8663	0.8745
	KL	0.2001	0.0611	0.3223	0.9155	0.8544	0.8503	0.8734
	GradDiff	0.1967	0.0670	0.3307	0.9122	0.8690	0.8553	0.8788
	DPO	0.1700	0.0627	0.2953	0.9085	0.8611	0.8518	0.8738
	NPO	0.1620	0.0671	0.2962	0.9164	0.8495	0.8402	0.8687
	RMU	0.1573	0.0601	0.2776	**0.9206**	0.8781	0.8650	**0.8879**
	CEL(Ours)	**0.1525**	**0.0341**	**0.2207**	0.9119	**0.8795**	**0.8670**	0.8861
Phi-1.5	GA	0.2588	0.0428	0.3445	0.7126	0.3974	0.7417	0.6172
	KL	0.3049	0.0416	0.3880	0.7231	0.4121	0.7407	0.6253
	GradDiff	0.3273	0.0438	0.4150	0.7394	0.4040	0.7417	0.6284
	DPO	0.2578	0.0438	0.3453	0.7442	0.4130	**0.7539**	0.6370
	NPO	0.2612	0.0423	0.3458	0.7456	0.4078	0.7472	0.6335
	RMU	0.2838	0.0372	0.3583	0.7386	**0.4163**	0.7468	0.6339
	CEL(Ours)	**0.2547**	**0.0252**	**0.3051**	**0.7552**	0.4119	0.7474	**0.6381**

Table 4: Unlearning results on the "forget01" split of TOFU dataset. S_{ED} is a weighted combination of S_{mean} and S_{std}. R_{avg} is the average of R_r, R_{RA}, and R_{WF}. The top two results in each column are highlighted in gray, and the best results are bold.

Model	Method	Unlearning Effect ↓			Model Utility ↑			
		S_{mean}	S_{std}	S_{ED}	R_r	R_{RA}	R_{WF}	R_{avg}
Llama-2-7B	GA	0.1205	0.0623	0.2451	0.5508	0.6821	0.6013	0.6114
	KL	0.1345	0.0721	0.2788	0.6468	0.7014	0.6582	0.6688
	GradDiff	0.1284	0.0610	0.2504	0.7893	0.8161	0.6241	0.7432
	DPO	0.1138	0.0707	0.2551	**0.8680**	**0.8332**	0.6550	0.7854
	NPO	0.1864	0.0693	0.3249	0.8230	0.7677	0.7151	0.7686
	RMU	0.1458	0.0625	0.2707	0.8530	0.8202	0.9065	0.8599
	CEL(Ours)	**0.0791**	**0.0365**	**0.1521**	0.8650	0.8204	**0.9210**	**0.8688**
Phi-1.5	GA	0.2096	0.0659	0.3414	0.3631	0.2829	0.6687	0.4382
	KL	0.2609	0.0706	0.4021	0.4287	0.3145	**0.7263**	0.4898
	GradDiff	0.3078	0.0635	0.4348	0.4702	0.2430	0.6766	0.4633
	DPO	0.2941	0.0740	0.4421	0.4948	0.3167	0.7097	0.5071
	NPO	0.2819	0.0779	0.4376	0.4641	**0.3663**	0.7051	0.5118
	RMU	**0.2120**	0.0614	0.3347	0.5963	0.3011	0.6343	0.5106
	CEL(Ours)	0.2382	**0.0404**	0.3189	**0.6357**	0.3096	0.6697	**0.5383**

Table 5: Unlearning results on the "forget10" split of TOFU dataset. S_{ED} is a weighted combination of S_{mean} and S_{std}. R_{avg} is the average of R_r, R_{RA}, and R_{WF}. The top two results in each column are highlighted in gray, and the best results are bold.

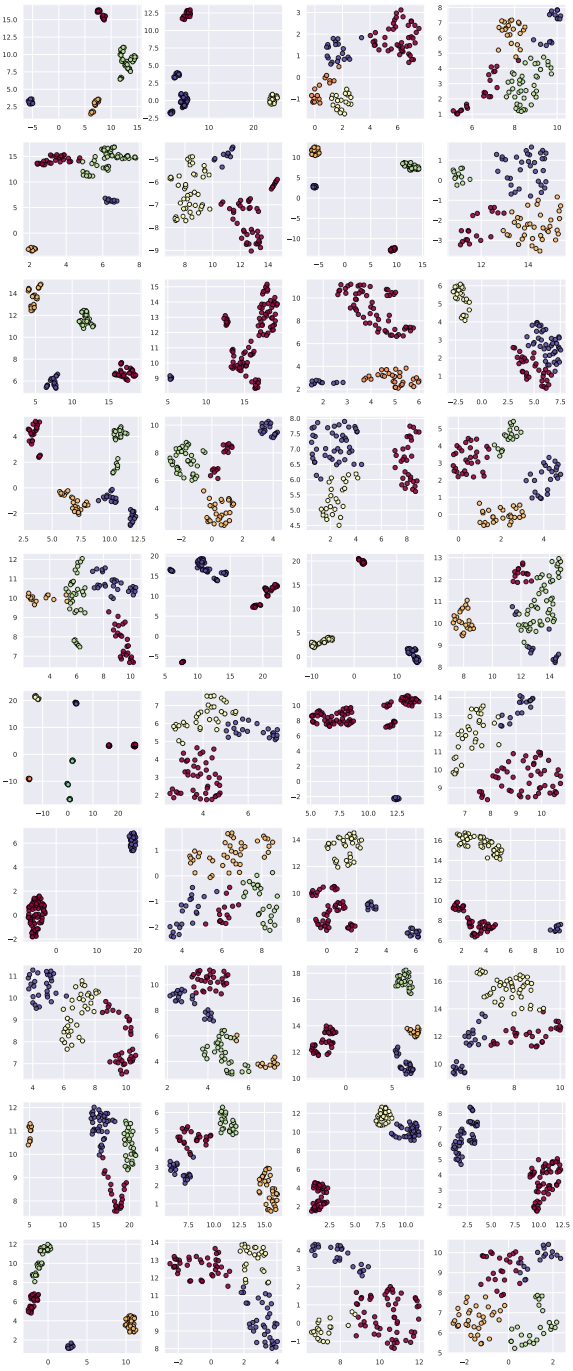

Figure 5: Clustering results on different S_q using Dirichlet Process Mixture Model on "forget05" split of TOFU. The number of cluster centers typically ranges in $2 \sim 4$.