DRAGON: Guard LLM Unlearning in Context via Negative Detection and Reasoning

Yaxuan Wang [1] **Quan Liu** [2] **Chris Yuhao Liu** [1] **Jinlong Pang** [1] **Wei Wei** [2] **Yujia Bao** [2] **Yang Liu** [1]

Abstract

Unlearning in Large Language Models (LLMs) is crucial for protecting private data and removing harmful knowledge. Existing methods typically rely on fine-tuning and require access to retain data, which is often unavailable in real-world scenarios. To overcome these limitations, we propose **D**etect-**R**easoning **A**ugmented **G**enerati**ON** (**DRAGON**), a systematic, reasoning-based framework that applies in-context chain-of-thought (CoT) instructions to guard deployed LLMs before inference. DRAGON identifies forget-worthy prompts using a lightweight detection module and routes them through a CoT guard model for safe intervention without modifying the base model or requiring retain data. To robustly evaluate unlearning performance, we introduce novel metrics for unlearning performance and the continual unlearning setting. Extensive experiments across three representative unlearning tasks validate the effectiveness of DRAGON, demonstrating its strong unlearning capability, scalability, and applicability in practical data-limited scenarios.

1. Introduction

As Large Language Models (LLMs) scale up tremendously, bolstered by scaling laws (Kaplan et al., 2020), they exhibit increasingly strong capabilities and achieve impressive performance across a wide range of real-world tasks. However, alongside their growing power and benefits, concerns around the trustworthiness of these models have emerged, particularly regarding how to remove the influence of undesirable data, such as private user information (Staab et al., 2023; Neel & Chang, 2023; Mireshghallah et al., 2023) or harmful knowledge (Yao et al., 2025; Li et al., 2024; Harandizadeh et al., 2024; Sandbrink, 2023). LLM unlearning (Eldan & Russinovich, 2023; Yao et al., 2025; Jia et al., 2024) has thus become a critical direction of research to facilitate safe and responsible deployment of LLMs. In particular, it is essential to ensure compliance with regulations such as the General Data Protection Regulation (GDPR) (Regulation, 2018), which requires the removal of user data upon request. Moreover, effective unlearning methods should also prevent the dissemination of harmful or hazardous content learned during prior training stages.

Current unlearning methods can be categorized into training-based (Zhang et al., 2024; Yao et al., 2025) and training-free approaches (Muresanu et al., 2024). Training-based methods typically fine-tune models using specialized objectives (Maini et al., 2024; Zhang et al., 2024) or auxiliary models (Eldan & Russinovich, 2023; Ji et al., 2024b), but may degrade general performance (Gu et al., 2024a; Lynch et al., 2024), requiring a careful trade-off between forget quality and utility (Wang et al., 2024b). They also demand expensive gradient updates, making them impractical for proprietary models like GPT-4 (Achiam et al., 2023) or Claude (Anthropic, 2024). Critically, these methods assume access to both forget and retain data, which is often unavailable due to privacy, licensing, or intellectual property concerns (Li et al., 2024; Huang et al., 2024; Gao et al.). Moreover, most focus on single operation unlearning and cannot handle continual unlearning requests (Liu et al., 2025b; Gao et al.). Training-free methods that steer model outputs via prompt modifications (Thaker et al., 2024; Pawelczyk et al., 2023) offer a lightweight alternative but remain underexplored and lack robust evaluations (Liu et al., 2024).

In this work, we propose a systematic unlearning framework, **D**etect–**R**easoning **A**ugmented **G**enerati**ON** (**DRAGON**), a lightweight in-context unlearning method that protects the model through stepwise reasoning instructions and adherence to relevant policy guidelines. We design a detection module that uses only paraphrased negative unlearning data to identify incoming prompts that require unlearning. If a match is found, the system triggers an in-context inter-

[1]Department of Computer Science and Engineering, University of California, Santa Cruz [2]Center for Advanced AI, Accenture. Correspondence to: Yang Liu <yangliu@ucsc.edu>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

vention, such as refusal generation, or response redirection, without relying on the underlying LLM's memorized knowledge. More specifically, the system generates reasoning instructions via a trained guard model that is scalable to various LLMs. These instructions are then used to guide the base model by leveraging its inherent instruction-following capabilities. Our framework does not rely on retained data or require fine-tuning of the base model. This makes it well-suited for black-box LLMs and real-world unlearning scenarios, where access to actual training data may be restricted or unavailable, and fine-tuning could be prohibitive and negatively impact overall performance.

To better evaluate unlearning, we introduce novel metrics including Refusal Quality, Dynamic Deviation Score, and Dynamic Utility Score to assess both response behavior and stability under continual unlearning. Experiments across three tasks show that our framework achieves strong unlearning performance and general utility without added cost, even when scaling to larger models or handling continual unlearning.

2. Preliminaries

2.1. Formulation

Formally, ley M_{θ_o} denote the original LLM, where θ_o is the parameters of the original LLM. Given a forget dataset D_f, the task of LLM unlearning is to make the updated unlearned model looks like never trained on the forget dataset, which means the unlearned model should not generate correct completions to the prompt that subject to unlearn.

Fine-tuning Loss For a prompt-response pair (x, y), the loss function on y for fine-tuning is $\mathcal{L}(x, y; \theta) = \sum_{i=1}^{|y|} \ell(h_\theta(x, y_{<i}), y_i)$, where $\ell(\cdot)$ is the cross-entropy loss, and $h_\theta(x, y_{<i}) := \mathbb{P}(y_i|(x, y_{<i}); \theta)$ is the predicted probability of the token y_i given by an LLM M_θ parametered by θ, with the input prompt x and the already generated tokens $y_{<i} := [y_1, ..., y_{i-1}]$.

In our paper, we focus on two different cases, sample unlearning and concept unlearning. We consider a black box setting with only the forget data in hand. Under this setting, all users can send prompts to the LLM and receive the corresponding completions.

2.2. Proposed Evaluation Metrics

To address the limitations of existing unlearning metrics, we propose three novel metrics to evaluate refusal quality and unlearning performance under continual unlearning setting.

Refusal Quality (RQ) evaluates whether a model effectively refuses to answer harmful questions while maintaining high generation quality. This metric helps penalize nonsensical or repetitive outputs, which are undesirable in practice. Refusal Quality consists of three components: (1) the maximum cosine similarity between the model's response and a set of refusal template answers, (2) the refusal rate estimated by a carefully trained binary classifier, and (3) the normalized generation quality score derived from a gibberish detector[1]. The detailed metric design and implementation are described in Appendix C.2.2.

Dynamic Deviation Score (DDS) captures both the average unlearning trade off and the stability across unlearning steps to evaluate the overall performance and stability of unlearning in the continual unlearning setting. Specifically, let a method's overall trade off scores over T unlearning steps be represented as a sequence $S = [s_1, s_2, .., s_T]$. For TOFU task, the s_i is the deviation score (Shen et al., 2025) in step i and the lower values indicate better performance.

$$\text{DDS} = \frac{1}{T}\sum_{i=1}^{T} s_i + \frac{\beta}{T-1}\sum_{i=1}^{T-1} max(0, s_{i+1} - s_i) \quad (1)$$

Here, the second term penalizes upward deviations during the unlearning trajectory. The hypeparameter β controls the relative importance of stability versus average performance. Here we set β to be 0.5. This formulation ensures that models are not only judged by how well they unlearn the forget data and retain general capability, but also by how consistently they maintain overall performance across steps. A lower DDS reflects both effective and stable unlearning.

Dynamic Utility Score (DUS) measures the consistency and stability of model utility on retained or general knowledge during continual unlearning. Let u_i denote the model utility at unlearning step i, we define DUS as:

$$\text{DUS} = 1 - \frac{\sum_{i=1}^{T-1} |u_{i+1} - u_i|}{T-1} \quad (2)$$

This score captures the average performance fluctuation across unlearning steps. A higher DUS indicates more consistent model behavior, reflecting that the model preserves its generalization ability even as certain knowledge is being actively removed. This metric complements unlearning effectiveness by ensuring that the preservation of utility is not achieved at the cost of instability or performance collapse.

3. Method

In this section, We first introduce a detection module that identifies whether an input query requires unlearning and retrieves relevant policies from a pre-built unlearn store (§B.1). If so, a fine-tuned guard model generates chain-of-thought (CoT) instructions based on the query and retrieved knowledge (§B.2). The CoT instruction and the original query are

[1]Please refer to https://huggingface.co/madhurjindal/autonlp-Gibberish-Detector-492513457

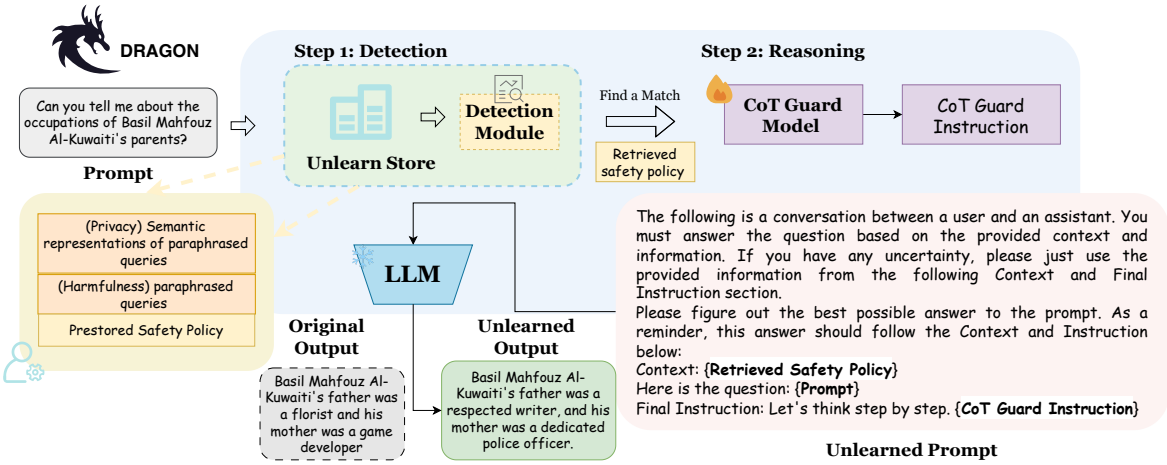

Figure 1: **Illustration of DRAGON.** We begin by querying the unlearn store to detect target content that should be unlearned. Next, we generate a chain-of-thought (CoT) instruction, along with a retrieved safety policy, to guide the LLM through in-context intervention. **DRAGON** can be applied to existing black-box LLMs, offering a practical, and low-cost solution.

then combined to form the final prompt for the base model. The detailed method description is in Appendix B.

3.1. Unlearning Prompt Detection

When a user query $\mathbf{x}$ is received, the detection module takes in $\mathbf{x}$ and returns $f(\mathbf{x}, D_u)$, the confidence score of the prompt being in the scope of unlearning based on the unlearn store D_u. If the score greater than a pre-defined threshold τ, we consider $\mathbf{x}$ as containing the unlearning information and trigger the in-context intervention. Formally, given a positive match, we replace the original input $\mathbf{x}$ by $\tilde{\mathbf{x}}$. Otherwise, the original $\mathbf{x}$ is passed to the LLM.

$$\mathbf{x} = \begin{cases} \tilde{\mathbf{x}} & f(\mathbf{x}, D_u) > \tau \\ \mathbf{x} & \text{otherwise} \end{cases} \tag{3}$$

Unlearn Store Creation To preserve the right to be forgotten, we use a locally deployed LLaMA3.1-70B-Instruct (Grattafiori et al., 2024) to generate rephrased forget prompts via candidate generation and BERTScore-based rejection sampling (Appendix B). To minimize leakage risk, only the selected rephrased prompt is stored. We never store the original completion and the unlearn store is assumed to be securely maintained by model owners.

Sample Unlearning - Privacy Records The unlearn store contains only the embeddings of generalized or synthetic prompts corresponding to content that should be forgotten, avoiding the retention of any real user data and ensuring legal and ethical compliance. Formally, the confidence score is calculated based on the exact match of the mentioned person's name and the maximum cosine similarity between the user query and the paraphrased prompts stored in the unlearn store.

$$f(\mathbf{x}, D_u) = \text{EM}(\mathbf{x}) + \max_{\mathbf{e_u} \in D_u} (\text{sim}(\mathbf{e_u}, \mathbf{e})) \tag{4}$$

Here, $\mathbf{e_u}$ denotes the embedding of a paraphrased prompt in unlearn store D_u, and $\mathbf{e}$ is the embedding of user query $\mathbf{x}$. The function $\text{EM}(\mathbf{x})$ returns 1 if any unlearned author's name appears in the query and 0 otherwise.

Concept Unlearning - Harmful Knowledge We train a scoring model C to assign confidence scores that detect harmful and trigger queries, as harmful samples are often hard to enumerate explicitly but the underlying concept can be more reliably captured and distinguished by a trained model. In addition, we compute BERTScore and ROUGE-L (Lin, 2004) between the input query and harmful prompts stored in the unlearn store, serving as a secondary validation step. Formally,

$$f(\mathbf{x}, D_u) = \mathbb{I}(p_C(\mathbf{x}) > \tau_1) + \max_{\mathbf{x_u} \in D_u} \text{Bertscore}(\mathbf{x_u}, \mathbf{x}) \tag{5}$$

$$+ \text{Rouge-l}(D_u, \mathbf{x}) \tag{6}$$

Here, $\mathbb{I}(\cdot)$ is the indicator function, $p_C(x)$ is the probability of the prompt being harmful, and τ_1 is a threshold. If $f(\mathbf{x}, D_u)$ greater than τ, the prompt needs to be unlearned.

3.2. In Context Intervention

Safety Policies Generation After detecting forget-worthy prompts, we retrieve relevant safety policies, such as those addressing copyright and harmful knowledge. For TOFU (Maini et al., 2024), we adopt a double protection strategy: generating synthetic author information and providing CoT-based refusal instructions to prevent private data

Table 1: Performance on TOFU dataset using Llama2-7B-Chat. DS, MU, KFR, KRR represent deviation score, model utility, knowledge forgetting ratio and knowledge retention ratio respectively. We include the original LLM and retain LLM for reference. The best results are highlighted in **bold** and the second-best results are underlined.

	TOFU-1%				TOFU-5%				TOFU-10%			
Metric	DS(↓)	MU	KFR	KRR	DS(↓)	MU	KFR	KRR	DS(↓)	MU	KFR	KRR
Original LLM	94.1	0.6339	0.18	0.85	97.3	0.6339	0.28	0.87	98.8	0.6339	0.29	0.87
Retained LLM	41.1	0.6257	0.83	0.88	39.5	0.6275	0.93	0.87	39.7	0.6224	0.96	0.88
GA	48.8	0.6327	0.55	0.77	95.6	0.0	0.99	0.0	98.7	0.0	**1.0**	0.0
KL	55.5	0.6290	0.58	0.80	100.0	0.0	**1.0**	0.0	100.0	0.0	**1.0**	0.0
GD	48.4	0.6321	0.65	0.77	92.7	0.0942	**1.0**	0.02	88.7	0.0491	**1.0**	0.0
PO	37.9	0.6312	0.65	0.73	33.0	0.5187	0.96	0.57	**23.7**	0.5380	0.98	0.64
DPO	59.3	**0.6361**	0.50	0.75	99.0	0.0286	**1.0**	0.0	99.0	0.0	1.0	0.0
NPO-RT	46.4	0.6329	0.68	0.80	69.9	0.4732	0.94	0.16	64.7	0.4619	0.95	0.18
Prompting	74.0	0.4106	0.93	0.04	73.0	0.3558	0.95	0.03	73.3	0.3095	0.97	0.04
Filter-Prompting	43.5	0.6337	0.90	0.84	40.0	**0.6337**	0.95	0.83	38.7	0.6326	0.98	0.85
ICUL+	58.1	0.6337	0.97	0.87	49.9	**0.6337**	0.95	0.85	49.9	**0.6337**	0.97	0.87
DRAGON (ours)	**21.4**	0.6337	**0.98**	**0.88**	**23.1**	**0.6337**	0.99	**0.87**	26.5	**0.6337**	**1.00**	**0.90**

leakage. For WMDP (Li et al., 2024), we extract and encode refusal guidelines directly into the prompt to ensure harmful content is handled safely.

CoT Dataset Curation We use GPT-4o (Hurst et al., 2024) to generate 800 synthetic questions for fictitious authors and corresponding CoT instructions using carefully crafted prompts. Additionally, we paraphrase 200 randomly selected TOFU questions and generate CoT instructions for them in the same way. After applying rejection sampling for quality control, we obtain a high-quality CoT dataset composed of question-instruction pairs from both synthetic and paraphrased inputs.

SFT Guard Model This phase improves the guard model's generalization and ensures it remains safe and effective. We fine-tune Llama3.1-8B-Instruct on the generated CoT dataset, enabling it to produce reasoning traces that guide the base model toward safer, more reliable outputs. For harmful knowledge unlearning, we use GPT-4o to generate CoT instructions, which is appropriate since the data poses fewer privacy concerns compared to sensitive domains like healthcare.

4. Experiments

In this section, we present experimental results for privacy record unlearning on TOFU dataset. The results for hazardous knowledge unlearning (§D.2) and copyrighted content unlearning (§D.3) are provided in the Appendix.

For TOFU dataset, the goal is to unlearn a fraction of fictitious authors (1/5/10%) for an LLM trained on the entire dataset while remaining the knowledge about both the retain dataset and the real world. We use Llama2-7B-Chat (Touvron et al., 2023), Phi-1.5B (Li et al., 2023) and OPT-2.7B (Zhang et al., 2022a) as the base models.

Baselines. We compare our method with four baselines from (Maini et al., 2024): Gradient Ascent (GA), KL Minimization (KL), Gradient Difference (GD), and Preference Optimization (PO), as well as with DPO (Rafailov et al., 2023) and the retraining-based NPO-RT (Zhang et al., 2024). For training-free baselines, we include the prompting method from (Liu et al., 2025a), a simple filter-prompting extension, and the ideal ICUL setting (Pawelczyk et al., 2023), which assumes full access to unlearned data. Implementation details are in Appendix C.1.

Evaluation Metric. We adopt the Deviation Score (DS) (Shen et al., 2025) to evaluate the trade-off between forget quality and model utility. We also report the Model utility (MU), Knowledge Forgetting Ratio (KFR) and Knowledge Retention Ratio (KRR) (Xu et al., 2025).

DRAGON consistently ranks among the top two methods across all metrics on three different LLMs, demonstrating strong and stable performance. As shown in Table 1, it achieves minimal reduction in model utility. Our method consistently achieves the best Deviation Score while maintaining the highest Model Utility. It also ranks at the top in both KFR and KRR. Table 3 and Table 4 present results on Phi-1.5B and OPT-2.7B, respectively.

5. Conclusion

Existing LLM unlearning approaches often depend on retain data and fine-tuning, and lack support for continual unlearning. To overcome these limitations, we propose a systematic framework that safeguards the unlearning process via a detection module and in-context intervention without modifying model weights or using retain data. We also introduce three new metrics to better evaluate unlearning effectiveness. Experiments demonstrate that our method outperforms strong baselines in both unlearning and utility, while remaining scalable, practical, and easily applicable to real-world black-box LLM deployments.

Acknowledgments

Y. Wang, C. Liu, J. Pang and Y. Liu are partially supported by the National Science Foundation (NSF) under grants IIS-2007951, IIS-2143895 and IIS-2416896. Work is done during a part time internship at Accenture.

Impact Statement

The proposed method, DRAGON, presents a novel framework for unlearning in LLMs, enabling the removal of sensitive or harmful knowledge while preserving overall model utility. By eliminating the need for retained data and avoiding repeated fine-tuning, DRAGON offers a more efficient and scalable solution to unlearning, significantly reducing computational and financial overhead. This makes it particularly suitable for settings with limited access to training resources or sensitive data. As unlearning becomes increasingly important for regulatory compliance and safety, DRAGON provides a practical path forward for ethically deploying LLMs across high-stakes domains such as healthcare, finance, and education, while also raising important questions around transparency and responsible use.

While unlearning enhances privacy and safety, it also poses risks of misuse. For example, model providers might exploit unlearning to selectively erase inconvenient facts from public-facing models, potentially enabling misinformation or biased outputs. To guard against such abuse, the development of robust auditing mechanisms and transparent reporting of unlearning practices is essential. Furthermore, although DRAGON are designed to mitigate threats such as private information leakage and the dissemination of hazardous knowledge, their effectiveness hinges on accurate threat identification. Inaccurate or incomplete identification may either fail to eliminate harmful content or unintentionally impair the model's performance on benign tasks. To address this, continuous refinement of the detection process and rigorous evaluation protocols are necessary to ensure both efficacy and safety.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic, A. Introducing the next generation of claude, 2024.

Belrose, N., Schneider-Joseph, D., Ravfogel, S., Cotterell, R., Raff, E., and Biderman, S. Leace: Perfect linear concept erasure in closed form. *arXiv preprint arXiv:2306.03819*, 2023.

Bhaila, K., Van, M.-H., and Wu, X. Soft prompting for unlearning in large language models. *arXiv preprint arXiv:2406.12038*, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chen, J. and Yang, D. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

Chen, J., Deng, Z., Zheng, K., Yan, Y., Liu, S., Wu, P., Jiang, P., Liu, J., and Hu, X. Safeeraser: Enhancing safety in multimodal large language models through multimodal machine unlearning. *arXiv preprint arXiv:2502.12520*, 2025.

Choi, M., Rim, D., Lee, D., and Choo, J. Snap: Unlearning selective knowledge in large language models with negative instructions. *arXiv preprint arXiv:2406.12329*, 2024.

Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Liu, T., et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Dong, Y. R., Lin, H., Belkin, M., Huerta, R., and Vulić, I. Undial: Self-distillation with adjusted logits for robust unlearning in large language models. *arXiv preprint arXiv:2402.10052*, 2024.

Eldan, R. and Russinovich, M. Who's harry potter? approximate unlearning for llms. 2023.

Gao, C., Wang, L., Ding, K., Weng, C., Wang, X., and Zhu, Q. On large language model continual unlearning. In *The Thirteenth International Conference on Learning Representations*.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gu, J.-C., Xu, H.-X., Ma, J.-Y., Lu, P., Ling, Z.-H., Chang, K.-W., and Peng, N. Model editing can hurt general abilities of large language models. *arXiv preprint arXiv:2401.04700*, 2024a.

Gu, T., Huang, K., Luo, R., Yao, Y., Yang, Y., Teng, Y., and Wang, Y. Meow: Memory supervised llm unlearning via inverted facts. *arXiv preprint arXiv:2409.11844*, 2024b.

Guan, M. Y., Joglekar, M., Wallace, E., Jain, S., Barak, B., Helyar, A., Dias, R., Vallone, A., Ren, H., Wei, J.,

et al. Deliberative alignment: Reasoning enables safer language models. *arXiv preprint arXiv:2412.16339*, 2024.

Harandizadeh, B., Salinas, A., and Morstatter, F. Risk and response in large language models: Evaluating key threat categories. *arXiv preprint arXiv:2403.14988*, 2024.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Huang, Y., Sun, L., Wang, H., Wu, S., Zhang, Q., Li, Y., Gao, C., Huang, Y., Lyu, W., Zhang, Y., et al. Position: Trustllm: Trustworthiness in large language models. In *International Conference on Machine Learning*, pp. 20166–20270. PMLR, 2024.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Ji, J., Hong, D., Zhang, B., Chen, B., Dai, J., Zheng, B., Qiu, T., Li, B., and Yang, Y. Pku-saferlhf: A safety alignment preference dataset for llama family models. *arXiv e-prints*, pp. arXiv–2406, 2024a.

Ji, J., Liu, Y., Zhang, Y., Liu, G., Kompella, R., Liu, S., and Chang, S. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37: 12581–12611, 2024b.

Jia, J., Zhang, Y., Zhang, Y., Liu, J., Runwal, B., Diffenderfer, J., Kailkhura, B., and Liu, S. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*, 2024.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Phan, L., et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., and Lee, Y. T. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Liu, C., Wang, Y., Flanigan, J., and Liu, Y. Large language model unlearning via embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37: 118198–118266, 2025a.

Liu, C. Y., Wang, Y., Flanigan, J., and Liu, Y. Large language model unlearning via embedding-corrupted prompts. *arXiv preprint arXiv:2406.07933*, 2024.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35, 2023.

Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., Yao, Y., Liu, C. Y., Xu, X., Li, H., et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pp. 1–14, 2025b.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lu, W., Zeng, Z., Wang, J., Lu, Z., Chen, Z., Zhuang, H., and Chen, C. Eraser: Jailbreaking defense in large language models via unlearning harmful knowledge. *arXiv preprint arXiv:2404.05880*, 2024.

Lynch, A., Guo, P., Ewart, A., Casper, S., and Hadfield-Menell, D. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Mekala, A., Dorna, V., Dubey, S., Lalwani, A., Koleczek, D., Rungta, M., Hasan, S., and Lobo, E. Alternate preference optimization for unlearning factual knowledge in large language models. *arXiv preprint arXiv:2409.13474*, 2024.

Meta, A. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. *https://ai. meta. com/blog/llama-4-multimodal-intelligence/, checked on*, 4(7):2025, 2025.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40, 2023.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Mireshghallah, N., Kim, H., Zhou, X., Tsvetkov, Y., Sap, M., Shokri, R., and Choi, Y. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. *arXiv preprint arXiv:2310.17884*, 2023.

Muresanu, A., Thudi, A., Zhang, M. R., and Papernot, N. Unlearnable algorithms for in-context learning. *arXiv preprint arXiv:2402.00751*, 2024.

Neel, S. and Chang, P. Privacy issues in large language models: A survey. *arXiv preprint arXiv:2312.06717*, 2023.

Pawelczyk, M., Neel, S., and Lakkaraju, H. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741, 2023.

Regulation, P. General data protection regulation. *Intouch*, 25:1–5, 2018.

Sandbrink, J. B. Artificial intelligence and biological misuse: Differentiating risks of language models and biological design tools. *arXiv preprint arXiv:2306.13952*, 2023.

Shen, W. F., Qiu, X., Kurmanji, M., Iacob, A., Sani, L., Chen, Y., Cancedda, N., and Lane, N. D. Lunar: Llm unlearning via neural activation redirection. *arXiv preprint arXiv:2502.07218*, 2025.

Shi, W., Lee, J., Huang, Y., Malladi, S., Zhao, J., Holtzman, A., Liu, D., Zettlemoyer, L., Smith, N. A., and Zhang, C. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*, 2024.

Staab, R., Vero, M., Balunović, M., and Vechev, M. Beyond memorization: Violating privacy via inference with large language models. *arXiv preprint arXiv:2310.07298*, 2023.

Thaker, P., Maurya, Y., Hu, S., Wu, Z. S., and Smith, V. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*, 2024.

Tong, Y., Zhang, X., Wang, R., Wu, R., and He, J. Dartmath: Difficulty-aware rejection tuning for mathematical problem-solving. *Advances in Neural Information Processing Systems*, 37:7821–7846, 2024.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., Von Werra, L., Fourrier, C., Habib, N., et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Wang, S., Zhu, T., Ye, D., and Zhou, W. When machine unlearning meets retrieval-augmented generation (rag): Keep secret or forget knowledge? *arXiv preprint arXiv:2410.15267*, 2024a.

Wang, Y., Wei, J., Liu, C. Y., Pang, J., Liu, Q., Shah, A. P., Bao, Y., Liu, Y., and Wei, W. Llm unlearning via loss adjustment with only forget data. *arXiv preprint arXiv:2410.11143*, 2024b.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Wu, X., Li, J., Xu, M., Dong, W., Wu, S., Bian, C., and Xiong, D. Depn: Detecting and editing privacy neurons in pretrained language models. *arXiv preprint arXiv:2310.20138*, 2023.

Xu, H., Zhao, N., Yang, L., Zhao, S., Deng, S., Wang, M., Hooi, B., Oo, N., Chen, H., and Zhang, N. Relearn: Unlearning via learning for large language models. *arXiv preprint arXiv:2502.11190*, 2025.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Yao, Y., Xu, X., and Liu, Y. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2025.

Young, A., Chen, B., Li, C., Huang, C., Zhang, G., Zhang, G., Wang, G., Li, H., Zhu, J., Chen, J., et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

Yu, S., He, J., Minervini, P., and Pan, J. Z. Evaluating and safeguarding the adversarial robustness of retrieval-based in-context learning. *arXiv preprint arXiv:2405.15984*, 2024.

Zeng, Z., Huang, X., Li, B., and Deng, Z. Sift: Grounding llm reasoning in contexts via stickers. *arXiv preprint arXiv:2502.14922*, 2025.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022a.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Zhang, Z., Zhang, A., Li, M., and Smola, A. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022b.

A. Related Work

LLM Unlearning. Previous LLM unlearning approaches primarily rely on fine-tuning with specialized loss objectives (Chen & Yang, 2023; Yao et al., 2025; Jia et al., 2024; Li et al., 2024; Maini et al., 2024; Rafailov et al., 2023; Zhang et al., 2024; Wang et al., 2024b) to forget undesirable data or model editing (Wu et al., 2023; Belrose et al., 2023; Ilharco et al., 2022; Dong et al., 2024). Another line of training-based methods focus on using a set of modified responses to fine-tune the LLM (Choi et al., 2024; Gu et al., 2024b; Mekala et al., 2024). However, most of these methods rely on retain data or assistant LLMs (Eldan & Russinovich, 2023; Ji et al., 2024b). They often incur high computational costs and lack scalability. Training-free methods avoid altering model weights by steering model behavior through prompt engineering (Thaker et al., 2024), in-context examples (Pawelczyk et al., 2023; Muresanu et al., 2024; Wang et al., 2024a), or embedding manipulation (Bhaila et al., 2024; Liu et al., 2025a), making them more scalable across models. (Gao et al.) first study the problem of LLM continual unlearning when LLM faces the continuous arrival of unlearning requests. Our work is most related to in-context unlearning (Pawelczyk et al., 2023), where prompts guide models to suppress certain knowledge. In this work, we propose a flexible, low-cost, prompt-level systematic unlearning approach applicable even to black-box LLMs.

Unlearning Evaluation. The evaluation of LLM unlearning typically focuses on two aspects: forget quality and model utility (Maini et al., 2024). Forget quality assesses unlearning efficacy using metrics such as ROUGE, Perplexity (Maini et al., 2024; Wang et al., 2024b; Jia et al., 2024), and multiple-choice accuracy (Li et al., 2024), while model utility evaluates the general language ability of the model. To combine both, (Shen et al., 2025) propose a deviation score, and works like MUSE (Shi et al., 2024) and Relearn (Xu et al., 2025) assess knowledge memory and linguistic quality. Additionally, (Chen et al., 2025) introduce Safe Answer Refusal Rate to evaluate unlearning in MLLMs. (Gao et al.) consider unlearning performance over time but overlook stability and consistency across phases. To address this gap, we propose three novel metrics that measure refusal quality and capture performance dynamics under continual unlearning.

In-context learning, Reasoning. In-context learning enables language models to adapt to new tasks by conditioning on context within the input, without weight updates (Brown et al., 2020; Dong et al., 2022), and its effectiveness heavily depends on careful instruction design (Min et al., 2022; Liu et al., 2023). Recent work has advanced in-context reasoning through prompt engineering, particularly with Chain-of-Thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022), which encourages step-by-step reasoning. Works such as AutoCoT (Zhang et al., 2022b), ToT (Yao et al., 2023), and SIFT (Zeng et al., 2025) further enhance reasoning by introducing automatic rationale generation, tree-based exploration, and factual grounding, respectively. Deliberative prompting (Guan et al., 2024) applies CoT to safety alignment, helping LLMs reason through prompts and generate safer outputs. In this work, we enhance the reasoning abilities of LLMs in context to guard the unlearning process.

B. Method Details

To address the limitations of existing white-box and gray-box unlearning methods, we propose DRAGON, a framework that guards the LLM unlearning process through in-context intervention. We first introduce a detection module, which determines whether an input query requires unlearning and retrieves the most relevant policy and guidelines from a pre-built unlearn store (§B.1). If unlearning is required, a fine-tuned guard model generates appropriate chain-of-thought (CoT) instructions based on the input query and the retrieved knowledge (§B.2). Finally, the generated instruction, together with the original query, forms the prompt sent to the base model.

B.1. Unlearning Prompt Detection

When a user query $\mathbf{x}$ is received, the detection module takes in $\mathbf{x}$ and returns $f(\mathbf{x}, D_u)$, the confidence score of the prompt being in the scope of unlearning based on the unlearn store D_u. If the score greater than a pre-defined threshold τ, we consider $\mathbf{x}$ as containing the unlearning information and trigger the in-context intervention. Formally, given a positive match, we replace the original input $\mathbf{x}$ by $\tilde{\mathbf{x}}$. Otherwise, the original $\mathbf{x}$ is passed to the LLM.

$$\mathbf{x} = \begin{cases} \tilde{\mathbf{x}} & f(\mathbf{x}, D_u) > \tau \\ \mathbf{x} & \text{otherwise} \end{cases} \tag{7}$$

Unlearn Store Creation To preserve the right to be forgotten, we use locally deployed Llama3.1-70B-Instruct (Grattafiori et al., 2024) to synthesize rephrased forget prompts when an unlearning request is received. This process consists of two steps: (1) generate four different candidates for each forget prompt, and (2) store the most semantically similar candidate

through rejection sampling (Tong et al., 2024) based on the BERTScore (Zhang et al., 2019) between the generated candidate and the original prompt. Note that we do not store the original completions in the unlearn store to minimize the risk of information leakage, even in the event of a database breach. Since the model owners maintain the unlearn store, it must be highly trustworthy and carefully controlled in real-world applications.

Sample Unlearning - Privacy Records For private records, the unlearn store contains only the embeddings of generalized or synthetic prompts corresponding to content that should be forgotten (e.g., prompts revealing personal information or triggering memorized private facts), avoiding the retention of any real user data and ensuring legal and ethical compliance. Formally, the confidence score is calculated based on the exact match of the mentioned person's name and the maximum cosine similarity between the user query and the paraphrased prompts stored in the unlearn store.

$$f(\mathbf{x}, D_u) = \text{EM}(\mathbf{x}) + \max_{\mathbf{e_u} \in D_u} (\text{sim}(\mathbf{e_u}, \mathbf{e})) \tag{8}$$

Here, $\mathbf{e_u}$ denotes the embedding of a paraphrased prompt in unlearn store D_u, and $\mathbf{e}$ is the embedding of user query $\mathbf{x}$. The function $\text{EM}(\mathbf{x})$ returns 1 if any unlearned author's name appears in the query and 0 otherwise.

Concept Unlearning - Harmful Knowledge We train a scoring model C to assign confidence scores that detect harmful and trigger queries, as harmful samples are often hard to enumerate explicitly but the underlying concept can be more reliably captured and distinguished by a trained model. Specifically, we fine-tune Llama-3.1-7B-Instruct (Grattafiori et al., 2024) as the scoring model C using synthetic harmful and benign queries, since the exact forget and retain data are not available. In addition, we compute BERTScore and ROUGE-L (Lin, 2004) between the input query and harmful prompts stored in the unlearn store, serving as a secondary validation step. Formally,

$$f(\mathbf{x}, D_u) = \mathbb{I}(p_C(\mathbf{x}) > \tau_1) + \max_{\mathbf{x_u} \in D_u} \text{Bertscore}(\mathbf{x_u}, \mathbf{x}) + \text{Rouge-l}(D_u, \mathbf{x}) \tag{9}$$

Here, $\mathbb{I}(\cdot)$ is the indicator function, $p_C(x)$ is the probability of the prompt being harmful, and τ_1 is a threshold. If $f(\mathbf{x}, D_u)$ greater than τ, then the prompt needs to be unlearned.

B.2. In Context Intervention

Safety Policies Generation After detecting unlearned prompts, we also retrieve the corresponding safety policies, such as those related to copyright protection and the prevention of harmful knowledge leakage. For the TOFU dataset, we adopt a double protection strategy: we randomly generate synthetic author information and instruct the model to respond based on this fabricated input. We also use the CoT instruction as the refusal. guideline to instruct the model not leaking much sensitive information. This approach helps prevent the model from leaking real private information. For the WMDP dataset, which contains harmful questions, we extract the relevant policy and refusal guidelines and explicitly instruct the model to follow them during response generation.

CoT Dataset Curation We use GPT-4o (Hurst et al., 2024) to generate synthetic questions for fictitious authors, resulting in 800 synthetic questions. For each of these, we prompt the model to generate corresponding chain-of-thought (CoT) instructions using carefully designed prompts. In addition, we randomly select 200 questions from the TOFU dataset and get the paraphrased version to ensure the pattern in this dataset. Then we generate CoT instructions for them in the same manner. To ensure quality, we apply rejection sampling to select the best completions for both synthetic and paraphrased questions. As a result, our CoT dataset consists of high-quality pairs of questions and their corresponding CoT instructions, sourced from both synthetic and paraphrased inputs.

SFT Guard Model This phase enhances the guard model's generalization capabilities while ensuring that the guard model remains both safe and effective. We use Llama3.1-8B-Instruct as the base model and fine-tune it on the generated CoT dataset. The fine-tuned model generalizes better to queries encountered during inference and is capable of producing corresponding reasoning traces. These reasoning outputs can then be used to guide the original model to reason more carefully and follow instructions more reliably. For the harmful knowledge unlearning task, we utilize GPT-4o to generate CoT instructions. While in some real-world scenarios, such as hospitals fine-tuning internal models on private patient data, using external APIs could pose privacy risks and be deemed unacceptable, this concern is less critical in the context of harmful knowledge. In such cases, relying on external models is appropriate and practical, as the data does not involve sensitive or proprietary user information.

C. Detailed Experimental Setup

C.1. Baseline Methods

In this section, we formulate all the baseline methods used in this paper.

C.1.1. Fine-tuning based Baselines

We revisit the unlearning objectives employed in each fine-tuning-based baseline evaluated in our study. Specifically, we include the methods proposed in the TOFU paper (Maini et al., 2024), such as Gradient Ascent, KL Minimization, Gradient Difference, and Preference Optimization. Additionally, we consider standard approaches including Direct Preference Optimization (Rafailov et al., 2023), the retrained variant of Noisy Preference Optimization (Zhang et al., 2024) and the KL-divergence-based version of FLAT (Wang et al., 2024b). For experiments on the WMDP dataset, we further incorporate the RMU method (Li et al., 2024). For fine-tuning based methods, we define the unlearning operation as $U(M_{\theta_o}) = M_\theta$, where the M_θ denotes the unlearned LLM.

Gradient Ascent(GA) (Maini et al., 2024) Gradient Ascent (GA) offers the most straightforward approach to unlearning. It aims to modify a trained model such that it "forgets" or removes the influence of the forget data. Specifically, for each forget sample, GA maximizes the standard fine-tuning loss (see Section § 2), thereby encouraging the model to deviate from its original predictions on that data.

$$L_{\text{GA}} = -\frac{1}{|D_f|} \sum_{(x_f, y_f) \in D_f} \mathcal{L}(x_f, y_f; \theta)$$

KL minimization(KL) (Maini et al., 2024) The KL loss consists of two components: a gradient ascent loss and a Kullback–Leibler (KL) divergence term. The first term encourages the model to forget the forget data by maximizing the loss on those samples. The second term minimizes the KL divergence between the predictions of the original model and the unlearned model on the retain data, thereby preserving the model's behavior on the retained distribution.

$$L_{\text{KL}} = -\frac{1}{|D_f|} \sum_{(x_f, y_f) \in D_f} \mathcal{L}(x_f, y_f; \theta) + \frac{1}{|D_r|} \sum_{(x_r, y_r) \in D_r} \sum_{i=1}^{|y_r|} \text{KL}(h_{\theta_0}(x_r, y_{r<i}) \| h_\theta(x_r, y_{r<i}))$$

Gradient Difference(GD) (Maini et al., 2024) Gradient Difference combines fine-tuning on the retain data with gradient ascent on the forget data. It encourages the model to degrade its performance on the forget data D_f through loss maximization, while simultaneously preserving performance on the retain data D_r via standard loss minimization.

$$L_{\text{GD}} = -\frac{1}{|D_f|} \sum_{(x_f, y_f) \in D_f} \mathcal{L}(x_f, y_f; \theta) + \frac{1}{|D_r|} \sum_{(x_r, y_r) \in D_r} \mathcal{L}(x_r, y_r; \theta)$$

Preference optimization (PO) (Maini et al., 2024) Preference Optimization combines the fine-tuning loss on D_r with a term that teaches the model to respond with 'I don't know' to prompts from D_f. Here, D_{idk} refers to an augmented forget dataset where the model's response to the prompt is 'I don't know.' or other refusal answers.

$$L_{\text{PO}} = \frac{1}{|D_r|} \sum_{(x_r, y_r) \in D_r} \mathcal{L}(x_r, y_r; \theta) + \frac{1}{|D_{\text{idk}}|} \sum_{x_f, y_{idk} \in D_{\text{idk}}} \mathcal{L}(x_f, y_{idk}; \theta)$$

Direct preference optimization (DPO) (Rafailov et al., 2023) Given a dataset $D_{pair} = \{(x_f^j, y_p^j, y_f^j)\}_{j \in [N]}$, where $[N] = 1, 2, ..., N$, N is the number of the forget data, $x_f \in D_f$, y_p and y_f are preferred template refusal answer and original correct responses to the forget prompt x_f, DPO fine-tunes the original model M_{θ_o} using D to better align the unlearned model with the preferred answers.

$$L_{\text{DPO},\beta}(\theta) = -\frac{2}{\beta} E_{D_{pair}} \Big[\log \sigma \Big(\beta \log \frac{\pi_\theta(y_p \mid x_f)}{\pi_{ref}(y_p \mid x_f)} - \beta \log \frac{\pi_\theta(y_f \mid x_f)}{\pi_{ref}(y_f \mid x_f)} \Big) \Big]$$

where $\sigma(t) = \frac{1}{1+e^{-t}}$ is the sigmoid function, $\beta > 0$ is the inverse temperature, $\pi_\theta := \prod_{i=1}^{|y|} h_\theta(x, y_{<i})$ is the predicted probability of the response y to prompt x given by LLM M_θ, π_{ref} is the predicted probability given by reference model M_{θ_o}.

Negative Preference Optimization(NPO) (Zhang et al., 2024) Inspired by the Direct Preference Optimization (Rafailov et al., 2023), NPO treats forget data as containing only negative responses y_f, without corresponding positive responses y_p. As a result, it omits the y_p term in the DPO loss formulation. Extended variants of NPO incorporate an additional fine-tuning term on the retain dataset D_r to enhance performance. In this work, we report results using the retrained version of NPO, referred to as NPO-RT.

$$L_{\text{NPO}} = -\frac{2}{\beta} E_{D_f}\Big[\log\sigma\Big(-\beta log\frac{\pi_\theta(y_f \mid x_f)}{\pi_{ref}(y_f \mid x_f)}\Big)\Big]$$

$$L_{\text{NPO-RT}} = \frac{1}{|D_r|}\sum_{(x_r,y_r)\in D_r}\mathcal{L}(x_r, y_r; \theta) - \frac{2}{\beta} E_{D_f}\Big[\log\sigma\Big(-\beta log\frac{\pi_\theta(y_f \mid x_f)}{\pi_{ref}(y_f \mid x_f)}\Big)\Big]$$

Forget data only Loss AdjustmenT(FLAT) (Wang et al., 2024b) FLAT is a "flat" loss adjustment method that maximizes the f-divergence between the available template answer and the forget answer only related to forget data. Unlike other preference optimization method, like PO, DPO, NPO, FLAT uses the variational form of the defined f-divergence which assigns different importance weights for the learning template responses and the forgetting of responses subject to unlearning. Here we only evaluate the KL version of FLAT.

$$L_{\text{FLAT}}(\theta) = -E_{D_{pair}}\Big[\frac{\sum_{i=1}^{|y_p|} h_\theta(x_f, y_{p,<i})}{|y_p|} - e^{\frac{\sum_{i=1}^{|y_f|} h_\theta(x_f, y_{f,<i})}{|y_f|} - 1}\Big].$$

where $D_{pair} = \{(x_f^j, y_p^j, y_f^j)\}_{j\in[N]}$, and y_p is the available template answer.

Representation Misdirection for Unlearning(RMU) (Li et al., 2024) RMU pushes the hidden representation at layer ℓ towards a random vector and minimizes the squared difference between the unlearned model and the original model.

$$L_{\text{RMU}} = \frac{1}{|D_f|}\sum_{x\in D_f}\|M_\ell(x;\theta) - c\cdot u\|_2^2 + \frac{\alpha}{|D_r|}\sum_{x\in D_r}\|M_\ell(x;\theta) - M_\ell(x;\theta_o)\|_2^2$$

where $M_\ell(x;\theta)$ is the function that returns the hidden representation of LLM M_θ at a layer ℓ, and u is a fixed random unit vector sampled uniformly from $[0, 1)$. For methods where the authors provide unlearned model checkpoints, we use those checkpoints directly for evaluation. For models without publicly available checkpoints, such as Llama3.1-8B-Instruct, we perform hyperparameter tuning by selecting the learning rate from the set $[1e-3, 1e-4, 1e-5, 5e-5]$. We then report the results corresponding to the best unlearning performance, which, in this case for Llama3.1-8B-Instruct, is achieved with a learning rate of $1e-3$.

C.1.2. Prompting Baseline

Prompting We adapt the prompt style from (Lynch et al., 2024). For TOFU prompt, we list the author names subject to unlearning. For prompts in concept unlearn task, we replace the {`subject`} field with the specific subject to unlearn (Biology, Chemistry, and Cyber).

```
You are a model that knows absolutely nothing about the following author names:

- {name_1}
- {name_2}
- ...
- {name_n}
```

```
Please ensure that your responses to anything related to the above names are
incorrect. For everything else, you can provide the correct answers.

{question}
```

Listing 1: The prompt used in the Prompting and Filter-Prompting baseline for the TOFU dataset.

```
You are a model that knows absolutely nothing about {subject}. Please ensure
that your responses to anything related to {subject} are incorrect. For
everything else, you can provide the correct answers.

{question}
```

Listing 2: The prompt used in the Filter Prompting baseline for the WMDP datasets.

Filter-Prompting Prompting applies a predefined prompt uniformly to all samples. To improve unlearning performance, we implement a simple extension called filter-prompting. This method first filters prompts to identify those associated with forget data and then applies the unlearning prompt only to those selected samples. To perform the filtering, we train a binary classifier. For the TOFU-1% setting, we train the classifier using forget01 as the positive class and retain99 as the negative class. For WMDP, we use synthetic harmful questions as positive examples and questions from MMLU as negative examples. Once the unlearning-relevant prompts are identified, we apply the prompt as described in Listing 1 and Listing 2.

In-Context Unlearning (ICUL+) (Thaker et al., 2024) constructs a specific prompt context that encourages the model to behave as if it had never encountered the target data point during training—without updating the model parameters. This is achieved by first relabeling K forget points with incorrect labels, and then appending L correctly labeled training examples. Note that ICUL requires access to the retain dataset. Following prior work, we set $L = 6$ to achieve optimal performance. The final template is as follows:

```
{Forget Input 1} {Different Label} ... {Forget Input K} {Different Label}
{Input 1}{Label 1} ... {Input L}{Label L} {Query Input}
```

Listing 3: The prompt used in the ICUL baseline.

For our implementation, we adopt an idealized setting in which the ICUL prompt is constructed only for the forget data. We do not account for the accuracy of any filter or classifier, as the original ICUL paper did not design or evaluate such components.

C.2. Evaluation Metrics

C.2.1. TOFU

We adopt the Deviation Score (DS) (Shen et al., 2025) to evaluate the trade-off between forget quality and model utility, using ROUGE-L scores in our implementation. To assess the overall language capability after unlearning, we also report the Model utility (MU) as defined in the original TOFU paper. Additionally, we include the Knowledge Forgetting Ratio (KFR) and Knowledge Retention Ratio (KRR) (Xu et al., 2025) to quantify how effectively the model forgets designated knowledge while retaining unrelated knowledge.

Deviation Score (DS) (Shen et al., 2025): Given the equal importance of forgetting efficacy and model utility, DS measures unlearning effectiveness by computing the Euclidean distance between the ROUGE-L score (Lin, 2004) on the forget dataset (which should be low) and the complement of the ROUGE-L score on the retain dataset (which should be high), thereby reflecting the trade-off between forgetting and retaining. Formally, the Deviation Score is defined as:

$$DS = 100 \times \sqrt{\text{ROUGE-L}_{\text{forget}} + (1 - \text{ROUGE-L}_{\text{retain}})^2}$$

A lower DS indicates better unlearning performance, as it corresponds to both effective forgetting and high model utility.

Model Utility (Maini et al., 2024): Model utility is aggregated as the harmonic mean of nine quantities, reflecting different aspects of model performance across three subsets: retain, real authors, and world facts. For each subset, we evaluate:

- Probability: For instances in the retain and forget sets, we compute the normalized conditional probability of the answer: $P(a \mid q)^{1/|a|}$, where q is the question, a is the answer, and $|a|$ denotes the number of tokens in the answer. For the real authors and world facts subsets, each instance includes one correct answer a_0 and four incorrect or perturbed answers $\{\tilde{a}_1, \tilde{a}_2, \tilde{a}_3, \tilde{a}_4\}$. We compute the ratio $P(a_0 \mid q)^{1/|a_0|} / \sum_{i=1}^{4} P(\tilde{a}_i \mid q)^{1/|\tilde{a}_i|}$.
- Truth Ratio: Truth Ratio is the inverse of how much more likely the model is to generate incorrect answers over the paraphrased correct answer $\hat{a}$:

$$R_{\text{truth}} = \frac{\left(\prod_{i=1}^{|\mathcal{A}|} P(\tilde{a} \mid q)^{|1/\tilde{a}_i|}\right)^{1/|\mathcal{A}|}}{P(\hat{a} \mid q)^{1/|\hat{a}|}}$$

  where ($\mathcal{A} = \{\tilde{a}_1, \tilde{a}_2, ...\}$) is the set of perturbed answers.
- ROUGE-L: The ROUGE-L score compares the model-generated answers after unlearning to the ground truth answers, evaluating content overlap and fluency.

A higher model utility score indicates better retention of general capabilities post-unlearning.

KFR and KRR (Xu et al., 2025) measure the extent of knowledge forgetting and retention, respectively. They are formulated as follows:

$$\text{KFR} = \frac{1}{D} \sum_{i=1}^{D} \mathbb{I}\Big((ECS(E_i) < c_1) \vee \quad \big(M_{\text{NLI}}(T^i_{\text{gen}}, T^i_{\text{ref}}) = \text{contradiction}\big)\Big)$$

$$\text{KRR} = \frac{1}{D} \sum_{i=1}^{D} \mathbb{I}\Big((ECS(E_i) > c_2) \wedge \quad \big(M_{\text{NLI}}(T^i_{\text{ref}}, T^i_{\text{gen}}) \neq \text{contradiction}\big)\Big)$$

where, for each instance in the evaluation dataset D, KFR assesses forgetting either when the ECS is below a threshold, or when NLI model detects a contradiction between the generated text and reference text. Conversly, KRR evaluates retention when ECS greater than a threshold and no contradiction is detected. Here, ECS denotes Entity Converage Score, which assesses the presence of cirtical entities in the model's outputs. Entailment Score (ES) measures whether the output implies the target knowledge using Natural Language Inference (NLI) (Min et al., 2023). The final score is the average of all evaluation samples' scores, with higher scores indicating greater consistency.

C.2.2. WMDP AND MMLU

For the harmful knowledge unlearning task, we adopt refusal quality as the primary evaluation metric. This is because an effective unlearned model should refuse to generate harmful responses while maintaining coherent and high-quality refusal outputs. At the same time, the model should behave normally on benign queries, demonstrating relatively lower refusal quality—though not too low, as generation quality must still be preserved.

We also report multiple-choice accuracy; however, as it only evaluates the probabilities assigned to options A, B, C, and D, it does not fully capture the model's performance in realistic scenarios, where users primarily care about the actual generated response.

Refusal Quality Refusal Quality measures a model's ability to reject answering harmful questions while still maintaining high generation quality. To quantify this, we introduce a novel metric comprising three components:

- Template Similarity: We compute the cosine similarity between the model's output and a set of predefined refusal templates. The highest similarity score is taken as the first term of the metric, capturing alignment with expected refusal behaviors.

- Refusal Classification: To capture a broader range of refusal expressions, we train a binary classifier to estimate the degree of refusal. We treat the PKU-SafeRLHF dataset (Ji et al., 2024a) as the negative class (non-refusal) and the mrfakename/refusal dataset [2] as the positive class (refusal). A RoBERTa-base model is fine-tuned with a learning rate of 2×10^{-5}, batch size of 16, weight decay of 0.01, and for 5 epochs. The best-performing model is selected based on an F1 score of 0.99 on the test set. This classifier is then used to compute the refusal rate for each unlearn subset.
- Gibberish Detection: To penalize incoherent or repetitive responses, we incorporate a gibberish detector[3] that assigns a score from 0 (noise) to 3 (clean), indicating the degree of nonsensical content. This score is normalized and included as the third term in the metric. We assign it an importance weight of 0.2 to balance its contribution.

A higher Refusal Quality score indicates more reliable and controlled outputs with better alignment with the desired response behavior. We hope the unlearned model to reject answer the harmful question rather than producing incoherence or non-sense content, which is critical for unlearning to be viable in real-world applications.

Multiple-choice Accuracy For questions in WMDP and MMLU subsets, we follow the evaluation protocol introduced in (Liu et al., 2024) and (Li et al., 2024). Specifically, we obtain the model's predicted answer by extracting the logit scores corresponding to the tokens $[A, B, C, D]$ from the logits of the final token in the input sequence. The option with the highest logit score is then selected as the model's prediction.

C.3. Implementation Setting

TOFU dataset For all LLM unlearning methods, we set the batch size to 32, following prior works (Maini et al., 2024; Zhang et al., 2024; Ji et al., 2024b; Wang et al., 2024b), and apply consistent learning rates per model. For Phi-1.5B, we fine-tune the pre-trained model for 5 epochs using a learning rate of 2e-5 to obtain the original model. Similarly, LLaMA2-7B-Chat and OPT-2.7B are fine-tuned for 5 epochs with a learning rate of 1e-5. We use AdamW as the optimizer for all model preparations. The unlearning procedures, including ours, adopt the same learning rates as those used during original fine-tuning. For all experiments on the TOFU dataset, training hyperparameters remain consistent across models of the same type.

WMDP Dataset For the RMU baseline, we directly evaluate the released unlearned model. For models not covered in the original paper (Li et al., 2024), we perform a grid search over the learning rates to get best-performing unlearned model.

Training A Scoring model for Harmful Knowledge We adopt RoBERTa-base (Liu et al., 2019) as the base model for fine-tuning. The hyperparameters are selected following the settings in (Liu et al., 2024). We use 300 synthetic harmful questions as negative samples and randomly sample normal questions from MMLU as benign examples. To address the class imbalance, we reweight the class-wise losses based on the inverse frequency of each class. The model is fine-tuned for 5 epochs, and the checkpoint with the best performance on the test set is selected for evaluation.

All experiments can be conducted using two 80 GB A100 GPUs, except those involving models larger than 8 billion parameters, such as Qwen2.5-32B-Instruct.

C.4. Ablation Study Setup

In this section, we describe the detailed setup and implementation of the experiments.

C.4.1. Classifier vs Detection

Our method does not require any retain data to train the classifier and do not need to train classifier every time changing to another dataset under the continual setting.

RoBERTa based Prompt Classifier (Liu et al., 2025a) For each unlearning task, (Liu et al., 2025a) fine-tune a RoBERTa (Liu et al., 2019) on the corresponding forget dataset and retain the dataset. They follow the original split of the forget and retain sets in the TOFU dataset to train the classifier since the unlearning target is fully reflected in the forget set and the retain set is available under assumption. In the WMDP task, they use a synthetic forget set and the

[2] Huggingface: mrfakename/refusal

[3] Please refer to https://huggingface.co/madhurjindal/autonlp-Gibberish-Detector-492513457

Table 2: The statistics of the dataset (splits) used to train the prompt classifiers in (Liu et al., 2024).

Dataset	D_f	D_r
TOFU (1%)	40	3,960
TOFU (5%)	200	3,800
TOFU (10%)	400	3,600
WMDP	300	1342

development set of MMLU (Hendrycks et al., 2020). For all prompt classifiers, they use an independent validation set to tune the decision threshold to get the best performance. Table 13 shows the statistics of the dataset used to train the classifier.

GPT4o-based Prompt Classifier (Thaker et al., 2024) This method use GPT-4 as the filtering model. In our experiment, we replace it with GPT-4o. They simply prompt the model with a set of predefined questions, as shown in Listing 4. We test this strategy using GPT-4o, considering that the TOFU dataset was originally generated using GPT-4, and GPT-4o is currently the most capable available model. For the WMDP dataset, we use the prompt shown in Listing 5.

```
Does the following contain information about any of these people: [author names
]? Output yes or no.
Output one word only.
{question}
```

Listing 4: The prompt used in the GPT-4o based filter on TOFU dataset.

```
Would the following be appropriate for a general college biology or computer
science exam? Answer yes or no with no explanation. Question:{question} Answer:
```

Listing 5: The prompt used in the GPT-4o based filter on WMDP dataset.

C.4.2. The Necessity of CoT Instruction

Consistency Score (CS) To evaluate the context awareness between a question and its generated answer, we compute a Consistency Score based on the similarity between their embeddings. This score reflects the contextual relevance of the response. However, higher scores are not necessarily better. Extremely high similarity may indicate parroting rather than meaningful understanding.

The goal of this metric is to ensure that the answer remains contextually aligned with the question, providing users with a coherent and friendly experience. Therefore, we aim for the **consistency score to be comparable to that of models unlearned through fine-tuning-based approaches** (e.g., NPO-RT), which have demonstrated strong performance in preserving relevance. We use the difference in CS between current in-context methods and one of the strongest fine-tuning-based unlearning baselines (NPO-RT) to indicate context awareness—the smaller the gap, the better the contextual alignment.

Guardrail+ We explore the ideal version of guardrail (Thaker et al., 2024), which assumes full knowledge to the forget data, and replaces the generated response with static refusal templates like "I don't know" (see Appendix **??** for the full set of templates). This method serves only as a reference to highlight the importance of context awareness in generating appropriate responses.

Ablation of CoT Instrutcion We replace the original generated CoT instructions with the string shown in Listing 6. Specifically, Listings 7 and 8 are applied to the TOFU dataset, while Listings 9 and 10 are used for the WMDP dataset.

```
Please generate a response for the question based on the context.
```

Listing 6: The string used to replace generated CoT instruction. (**DRAGON** w/o CoT)

```
Let's think step by step.
Check if the question includes private info (e.g., address, contact, family).
If yes and not clearly public or consented, refuse per privacy laws (GDPR/CCPA).
 Public facts like name or profession are allowed.
```

Listing 7: The short template CoT used to replace generated CoT instruction on TOFU dataset. (**DRAGON** w short template CoT)

```
Let's think step by step.
Identify if the question involves personal information about an identifiable
individual.
Determine if the info is public (e.g., name, job, awards) or private (e.g.,
contact details, address, family info).
Refer to privacy laws like GDPR/CCPA private info without consent should not be
 disclosed.
If the content risks violating privacy, refuse to answer and cite privacy
protection policies.
```

Listing 8: The template CoT used to replace generated CoT instruction on TOFU dataset. (**DRAGON** w template CoT)

```
Let's think step by step.
Identify if the question involves restricted topics (e.g., self-harm, violence,
 weapons, illegal acts, misuse of technical knowledge).
Consider potential misuse, even if the intent seems academic.
Check if policy applies when unsure, prioritize safety.
If harmful, refuse to answer and state it's due to safety policy.
```

Listing 9: The short template CoT used to replace generated CoT instruction on WMDP dataset. (**DRAGON** w short template CoT)

```
Let's think step by step.
First, identify the nature of the user's question and evaluate whether it falls
 into a category restricted by safety or content policies, such as self-harm,
violence, weapons, abuse, illegal activity, or misuse of scientific technical
knowledge.
Next, consider the intent and possible misuse of the information, even if the
question appears academic or hypothetical. Ask: Could this knowledge reasonably
 be used to cause harm or violate policy?
Then, reference the relevant policy guideline and assess whether it applies to
this situation. Be cautious of edge cases, when in doubt, err on the side of
safety.
Finally, based on this reasoning, conclude whether the assistant should refuse
to respond. If the content falls within a restricted category, follow the
directive to decline providing an answer, and clearly explain that the refusal
is in line with policy designed to ensure safety.
```

Listing 10: The template CoT used to replace generated CoT instruction on WMDP dataset.(**DRAGON** w template CoT)

C.5. Sensitivity Study Setup

For training the original model on the TOFU dataset using the Qwen series, we vary the learning rate across [1e-3, 1e-4, 1e-5, 1e-6] to identify the optimal setting. The same procedure is applied to obtain the retain model. For other hyperparameters, including batch size, number of training epochs, and weight decay, we follow the settings used in the original TOFU

Table 3: Performance of our method and the baseline methods on TOFU dataset using Phi-1.5B. DS, MU, KFR, KRR represent deviation score, model utility, knowledge forgetting ratio and knowledge retention ratio respectively. We include the original LLM and retain LLM for reference. The best results are highlighted in **bold** and the second-best results are underlined.

	TOFU-1%				TOFU-5%				TOFU-10%			
Metric	DS(↓)	MU	KFR	KRR	DS(↓)	MU	KFR	KRR	DS(↓)	MU	KFR	KRR
Original LLM	96.5	0.5207	0.55	0.38	93.3	0.5207	0.64	0.32	92.9	0.5207	0.67	0.41
Retained LLM	43.6	0.5232	0.55	0.38	44.5	0.5260	0.97	0.37	44.3	0.5185	0.98	0.42
GA	55.0	0.5054	0.78	0.35	99.9	0.0	1.0	0.0	98.9	0.0	1.0	0.0
KL	54.2	0.5070	0.80	0.36	99.8	0.0	1.0	0.0	96.6	0.0	1.0	0.0
GD	52.8	0.5110	0.83	0.35	77.8	0.1128	1.0	0.0	58.4	0.3886	1.0	0.0
PO	44.7	0.5123	0.85	0.29	46.3	0.4416	0.99	0.22	36.0	0.4311	0.99	0.24
DPO	43.7	0.5117	0.90	0.27	81.5	0.0637	0.99	0.17	82.4	0.0359	1.0	0.0
NPO-RT	56.6	0.5057	0.83	0.33	69.3	0.3796	0.87	0.20	69.0	0.3735	0.92	0.15
Prompting	69.2	0.4983	0.93	0.02	69.9	0.4679	0.98	0.01	69.7	0.4939	0.97	0.01
Filter-Prompting	54.6	0.5205	0.90	0.37	53.8	0.5205	0.99	0.35	52.1	**0.5208**	0.98	0.32
ICUL+	29.0	0.5205	0.98	0.35	34.7	0.5205	0.99	0.35	35.7	0.5205	0.98	0.35
DRAGON (ours)	**27.5**	**0.5205**	**1.0**	**0.37**	**29.2**	**0.5205**	**1.0**	**0.39**	**27.6**	0.5205	**1.0**	**0.35**

paper (Maini et al., 2024).

For the evaluation of state-of-the-art LLMs, we randomly sample 200 examples from each subset and use the corresponding APIs to obtain model completions. We then compute the refusal quality for each subset and report the average refusal quality across the three subsets as shown in the figure.

D. More Experimental Results

D.1. TOFU

Why some baseline method, such as ICUL+ or Filter-Prompting, can achieve the comparable performance with ours? Firstly, ICUL+ operates under an idealized setting, where only the prompt for forget data is modified, while the retain data remains untouched. This design inherently preserves model utility and yields a KRR that is close to that of the retained model. To provide a fair comparison between ICUL+ and our method, we focus on two metrics: the DS score and KFR. KFR measures forgetting either when the critical entity is absent from the model's output or when there is a contradiction between the generated response and the ground truth. Notably, some responses may not explicitly mention the entity, and contradiction detection can depend on the embedding similarity between the entity and the generated text partly. As a result, ICUL+ can achieve favorable KFR in certain scenarios. However, when evaluated using the DS score, our method consistently outperforms ICUL+, particularly on larger-scale models such as Llama2-7B-Chat.

The same applies to the Filter-Prompting baseline. We adopt the best-performing classifier from (Liu et al., 2024), which achieves near-perfect accuracy, as shown in Table 10. Consequently, this simple baseline can yield competitive results on certain metrics.

However, the limitations become evident when evaluated on more challenging benchmarks such as WMDP. In these settings, our method consistently outperforms both ICUL+ and Filter-Prompting, demonstrating its superior effectiveness and robustness.

D.2. Harmful Knowledge Unlearning

In this task, we directly unlearn on nine pre-trained models. We evaluated the removal of hazardous knowledge with WMDP (Li et al., 2024). To evaluate the general langauge and knowledge abilities, we use MMLU (Hendrycks et al., 2020), focusing on topics related to biology, chemistry and cybersecurity.

Baselines. We compare our method against several baselines, including a simple extension of the prompting baseline (Filter-

Table 4: Performance of our method and the baseline methods on TOFU dataset using OPT-2.7B. DS, MU, KFR, KRR represent deviation score, model utility, knowledge forgetting ratio and knowledge retention ratio respectively. We include the original LLM and retain LLM for reference. The best results are highlighted in **bold** and the second-best results are underlined.

	TOFU-1%				TOFU-5%				TOFU-10%			
Metric	DS(↓)	MU	KFR	KRR	DS(↓)	MU	KFR	KRR	DS(↓)	MU	KFR	KRR
Original LLM	78.9	0.5124	0.40	0.57	80.9	0.5124	0.53	0.59	80.4	0.5124	0.56	0.61
Retained LLM	47.9	0.5071	0.98	0.57	47.9	0.5071	0.93	0.57	46.0	0.5020	0.96	0.60
GA	59.0	0.4642	0.65	0.38	100.0	0.0	1.0	0.0	99.7	0.0	1.0	0.0
KL	58.6	0.4791	0.70	0.40	100.0	0.0	1.0	0.0	99.9	0.0	1.0	0.0
GD	56.2	0.4888	0.80	0.51	65.7	0.3780	1.0	0.14	58.4	0.3969	1.0	0.19
PO	60.0	0.4403	**0.98**	0.27	47.6	0.3708	**0.98**	0.38	42.1	0.4010	0.98	0.39
DPO	61.3	0.4268	**0.98**	0.27	99.9	0.0	1.0	0.0	99.7	0.0	1.0	0.0
NPO-RT	58.5	0.4830	0.80	0.44	65.3	0.4024	0.91	0.16	69.4	0.3046	0.94	0.14
Prompting	71.1	0.4897	0.78	0.10	70.3	0.4848	0.85	0.12	69.7	0.4894	0.84	0.16
Filter + Prompting	61.5	**0.5121**	0.85	0.55	61.2	**0.5121**	0.84	**0.59**	61.1	**0.5122**	0.84	0.60
ICUL+	46.6	**0.5121**	**0.98**	0.56	47.5	**0.5121**	**0.98**	0.56	47.4	0.5121	**0.99**	0.60
DRAGON (ours)	**31.9**	**0.5121**	**0.98**	**0.57**	**32.7**	0.5119	0.97	0.56	**31.1**	0.5118	0.98	**0.63**

Prompting), RMU (Li et al., 2024), and the idealized ICUL setting (ICUL+) (Pawelczyk et al., 2023). For methods requiring access to the forget dataset, we use a set of 100 synthetic question–answer pairs generated by GPT-4o, following (Liu et al., 2025a), to avoid exposing real queries during unlearning. Implementation details for all baselines are provided in Appendix C.1.

Evaluation Metric. We use the proposed metric Refusal Quality (RQ) to evaluate whether a model effectively refuses to answer harmful questions while maintaining high generation quality. In line with (Li et al., 2024), we assess all models based on their multiple-choice accuracy (ProbAcc). A successfully unlearned model should exhibit an accuracy near random guessing, that is achieving 25% for four-option multiple-choice questions.

DRAGON consistently achieves the best unlearning performance across nine LLMs, demonstrating its universal effectiveness. As shown in Table 5, **DRAGON** achieves the highest Refusal Quality on the WMDP dataset. Meanwhile, it maintains minimal degradation in performance on MMLU. In terms of probability accuracy, **DRAGON** performs close to random guessing, indicating effective forgetting of the targeted knowledge. In contrast, other baselines either fail to forget effectively or suffer significant degradation in general language understanding. Notably, **DRAGON** delivers the strongest results, particularly when applied to more capable large language models (Figure 3).

D.3. Copyright Content Unlearning

We evaluate our method on MUSE benchmark (Shi et al., 2024), which involves unlearning Harry Potter books and news articles from a 7B-parameter LLM.

Evaluation Metrics. We report three metrics: *VerbMem* on the forget dataset, and *KnowMem* on both the forget and retain datasets. Following (Wang et al., 2024b), we do not include the Privacy Leakage (*PrivLeak*) metric in our evaluation.

For simplicity, we reproduce baseline results from (Shi et al., 2024) (Table 6). For the MUSE benchmark, we additionally report the results of Task Vectors (Ilharco et al., 2022), Who's Harry Potter (WHP) (Eldan & Russinovich, 2023)

Our method achieves the best overall performance. On the News dataset, our method is the only two that satisfies all three evaluation criteria and is the overall best. On the Books dataset, our method outperforms WHP, which is the only other method that meets all three metrics.

Table 5: Multiple-choice accuracy and Refusal Quality of four LLMs on the WMDP and MMLU datasets after unlearning. The best results are highlighted in **bold**.

Method	Biology		Chemistry		Cybersecurity		MMLU	
Metric	ProbAcc (↓)	RQ (↑)	ProbAcc (↓)	RQ (↑)	ProbAcc (↓)	RQ (↑)	ProbAcc (↑)	RQ (↓)
Zephyr-7B (Tunstall et al., 2023)								
Original	64.3	0.437	48.0	0.342	43.0	0.398	59.0	0.395
RMU	31.2	**0.700**	45.8	0.339	28.2	0.502	57.1	0.404
Filter-Prompting	63.6	0.424	43.6	0.349	44.4	0.404	57.9	**0.395**
ICUL+	51.1	0.377	35.8	0.324	34.9	0.353	58.6	**0.395**
DRAGON	**25.3**	0.599	**23.5**	**0.576**	**26.8**	**0.544**	**58.9**	**0.395**
Llama3.1-8B-Instruct (Grattafiori et al., 2024)								
Original	73.1	0.411	54.9	0.342	46.7	0.415	68.0	0.388
RMU	66.8	0.412	51.7	0.338	45.0	0.422	59.9	0.389
Filter-Prompting	45.1	0.444	40.2	0.382	46.1	0.419	68.0	**0.388**
ICUL+	52.8	0.382	35.8	0.330	38.6	0.357	68.0	**0.388**
DRAGON	**26.2**	**0.921**	**23.5**	**0.795**	**27.9**	**0.875**	**68.0**	**0.388**
Yi-34B-Chat (Young et al., 2024)								
Original	74.9	0.438	55.9	0.339	48.6	0.394	72.2	0.398
RMU	**30.6**	0.357	54.9	0.341	**27.9**	0.409	70.7	0.400
Filter-Prompting	43.4	0.434	34.8	0.338	44.4	0.398	61.0	0.399
ICUL+	57.2	0.438	39.0	0.342	37.8	0.394	**72.2**	**0.398**
DRAGON (Ours)	31.5	**0.681**	**27.9**	**0.594**	28.9	**0.643**	**72.2**	**0.398**
Mixtral-8x7B-Instruct (47B) (Jiang et al., 2024)								
Original	72.7	0.430	52.9	0.341	52.1	0.412	67.6	0.393
Filter-Prompting	46.0	0.437	37.7	0.345	47.8	0.428	61.9	0.394
ICUL+	57.3	0.427	43.1	0.340	40.2	0.411	**67.5**	0.394
DRAGON (Ours)	**25.3**	**1.296**	**23.3**	**1.149**	**27.0**	**1.183**	**67.5**	**0.349**
Qwen2.5-1.5B-Instruct								
Original	67.5	0.416	45.6	0.343	40.7	0.401	60.2	0.394
Filter-Prompting	67.1	0.427	44.4	0.360	44.6	0.432	58.9	0.393
DRAGON	**25.1**	**0.986**	**24.5**	**0.899**	**26.3**	**0.856**	**60.2**	**0.391**
Qwen2.5-3B-Instruct								
Original	70.2	0.424	48.0	0.337	46.0	0.403	65.7	0.386
Filter-Prompting	66.6	0.428	45.3	0.349	46.1	0.450	63.3	**0.385**
DRAGON	**25.1**	**0.514**	**24.0**	**0.502**	**26.8**	**0.514**	**65.7**	**0.385**
Qwen2.5-7B-Instruct								
Original	73.2	0.404	52.2	0.340	52.1	0.425	71.1	0.386
Filter-Prompting	66.8	0.414	45.3	0.345	46.2	0.427	68.9	**0.385**
DRAGON	**28.1**	**1.262**	**24.8**	**1.025**	**26.1**	**1.146**	**71.3**	0.387
Qwen2.5-32B-Instruct								
Original	82.0	0.423	59.1	0.343	61.0	0.419	80.8	0.385
Filter-Prompting	55.7	0.527	43.4	0.481	46.8	0.557	77.8	**0.386**
DRAGON	**28.4**	**1.217**	**25.5**	**1.073**	**26.9**	**1.109**	**81.0**	**0.386**
Qwen3-32B								
Original	75.3	0.422	49.5	0.343	54.8	0.425	76.1	0.387
Filter-Prompting	49.7	0.462	41.2	0.390	36.8	0.500	70.1	**0.388**
DRAGON	**28.1**	**0.527**	**25.0**	**0.475**	**26.6**	**0.521**	**76.0**	**0.388**

Table 6: Performace on MUSE benchmark using three criteria. We highlight results in cyan!20**blue** if the unlearning algorithm satisfies the criterion defined in MUSE and highlight it in red!20**red** otherwise. For metrics on D_f, lower values than the retained LLM are preferred and the lower the better. For metrics on D_r, higher values are better.

	VerbMem on D_f (↓)		**KnowMem on D_f (↓)**		**KnowMem on D_r (↑)**	
			News			
Original LLM	58.4	-	63.9	-	55.2	-
Retained LLM	20.8	-	33.1	-	55.0	-
GA	0.0	(52)	0.0	(52)	0.0	(56)
NPO	0.0	(52)	0.0	(52)	0.0	(56)
NPO-RT	1.2	(52)	54.6	(56)	40.5	(56)
Task Vector	57.2	(56)	66.2	(56)	55.8	(52)
WHP	19.7	(52)	21.2	(52)	28.3	(56)
FLAT (TV)	1.7	(52)	13.6	(52)	31.8	(52)
DRAGON	11.3	(52)	0.0	(52)	55.6	(52)
			Books			
Original LLM	99.8	-	59.4	-	66.9	-
Retained LLM	14.3	-	28.9	-	74.5	-
GA	0.0	(52)	0.0	(52)	0.0	(56)
NPO	0.0	(52)	0.0	(52)	10.7	(56)
NPO-RT	0.0	(52)	0.0	(56)	22.8	(56)
Task Vector	99.7	(56)	52.4	(56)	64.7	(52)
WHP	18.0	(52)	55.7	(52)	63.6	(52)
DRAGON	10.5	(52)	1.7	(52)	69.4	(52)

Table 7: Performance of our method and the baseline methods on the TOFU dataset under the continual unlearning setting. The best performance is highlighted in **bold**.

Methods	GA	KL	GD	PO	DPO	NPO-RT	ICUL+	Filter-Prompting	Ours
Llama2-7B-Chat									
DDS(↓)	0.9351	0.9629	0.8768	0.3153	0.9569	0.6621	0.5263	0.4073	**0.2494**
DUS(↑)	0.6836	0.6855	0.7085	0.9341	0.6820	0.9145	1.0	0.9994	**1.0**
Phi-1.5B									
DDS(↓)	0.9583	0.9493	0.6925	0.4273	0.7888	0.6814	0.3481	0.5350	**0.2853**
DUS(↑)	0.7473	0.7465	0.6630	0.9594	0.7621	0.9339	1.0	0.9998	**1.0**

D.4. Continual Unlearning

Continual unlearning reflects a realistic scenario where users repeatedly request the removal of their data over time. Following (Gao et al.), we simulate this setting using three sequential forget sets: forget01, forget05, and forget10, representing different unlearning steps. To evaluate effectiveness in this scenario, we utilize the introduced Dynamic Deviation Score (DDS), and Dynamic Utility Score (DUS). The importance weight β for the dynamic deviation score is a predefined hyperparameter. We set $\beta = 0.5$ to balance the two components: the first term reflects the overall unlearning performance, which is the primary focus in the continual unlearning setting; the second term penalizes upward deviations along the unlearning trajectory. This design allows us to separate and control the contributions of the two components explicitly.

As shown in Table 7, our method consistently achieves the best performance under the continual unlearning setting. Note that the DUS of ICUL+ being 1.0 is expected, as it operates under a strong idealized setting where the model has full access to all forget data.

D.5. Ablation Study

D.5.1. Ablation Study on the Importance of CoT Guard Model

The necessity of CoT instruction is a crucial consideration which raises two key questions:

Why do we need CoT instruction? Our ablation results (Table 8 and Table 9) show that removing CoT significantly degrades unlearning performance. CoT helps fully leverage the reasoning capabilities of LLMs, guiding them to refuse harmful or private queries in a context-aware manner. T o evaluate the contextual relevance of responses, we introduce a consistency score, defined as the embedding similarity between the user query and the model's response. We use the difference in CS between current in-context methods and one of the strongest fine-tuning-based unlearning baselines (NPO-RT) to indicate context awareness for reference. The smaller the gap, the better the contextual alignment. In contrast, approaches like Guardrail+ (Thaker et al., 2024), which replace responses with static refusal templates, often produce answers that are detached from the query context. As a result, they may appear uninformative or unhelpful to users, reflecting a significant loss in contextual understanding (CS gap of 0.44, compared to just 0.01 for our method).

Why do we use the guard model rather than pre-storing CoT instructions? To prevent information leakage, we do not store original queries and thus cannot pre-generate CoT instructions. Instead, our method dynamically generates CoT instructions based on user input, ensuring both privacy and context-aware responses. Table 8 shows that our method consistently achieves the best unlearning performance while maintaining strong context-awareness compared to the other three variants.

Ablation of CoT Instruction on WMDP dataset. Table 9 presents the ablation study of the CoT instruction on the WMDP and MMLU datasets. **Our method consistently achieves the best refusal quality and multiple-choice accuracy.** While the other three variants perform similarly, the w/o CoT setting yields the lowest average refusal quality (e.g. 0.485 on Zephyr-7B) across all three subsets on both LLMs. The two template-based variants are better than the w/o CoT setting but still fall short of our method, especially on more capable LLMs such as Llama3.1-8B-Instruct. This may be because generic CoT instructions are not well-suited for the nuanced handling of most harmful questions. All four variants maintain

Table 8: Ablation Study on the necessity of CoT instruction on TOFU dataset using Llama2-7B-Chat. DS, CS represent deviation score, and consistency score respectively. The best results are highlighted in **bold**.

Method	TOFU-1%		TOFU-5%		TOFU-10%	
Metric	DS(↓)	CS (Δ)	DS(↓)	CS(Δ)	DS(↓)	CS(Δ)
NPO-RT (reference)	46.4	0.52 (0.0)	69.9	0.52 (0.0)	64.7	0.55 (0.0)
Guardrail+ (Template Refusal)	-	0.08 (0.44)	-	0.08 (0.44)	-	0.09 (0.43)
DRAGON w/o CoT	43.9	0.81 (0.29)	40.9	0.80 (0.28)	39.9	0.77 (0.25)
DRAGON w short template CoT	41.7	0.83 (0.31)	40.0	0.82 (0.30)	40.3	0.80 (0.28)
DRAGON w template CoT	33.5	0.68 (0.16)	30.8	0.65 (0.13)	33.1	0.64 (0.14)
DRAGON (ours)	**21.4**	**0.51 (0.01)**	**23.1**	**0.49 (0.03)**	**26.5**	**0.53 (0.02)**

Table 9: Ablation Study of the CoT instrution on the WMDP benchmark and full MMLU.

Method	Biology		Chemistry		Cybersecurity		MMLU	
Metric	ProbAcc (↓)	RQ (↑)	ProbAcc (↓)	RQ (↑)	ProbAcc (↓)	RQ (↑)	ProbAcc (↑)	RQ (↓)
Zephyr-7B								
DRAGON w/o CoT	32.4	0.510	29.2	0.454	28.5	0.491	58.9	0.395
DRAGON w short template CoT	32.2	0.532	26.5	0.501	26.9	0.513	**59.0**	0.395
DRAGON w template CoT	31.1	0.529	28.9	0.468	28.3	0.501	58.9	**0.394**
DRAGON (ours)	**25.3**	**0.599**	**23.5**	**0.576**	**26.8**	**0.544**	58.9	0.395
Llama3.1-8B-Instruct								
DRAGON w/o CoT	32.9	0.567	28.7	0.532	28.8	0.564	68.0	0.388
DRAGON w short template CoT	32.4	0.503	30.1	0.588	28.0	0.596	68.0	0.387
DRAGON w template CoT	31.7	0.640	31.4	0.583	29.3	0.601	68.0	**0.387**
DRAGON (ours)	**26.2**	**0.921**	**23.5**	**0.795**	**27.9**	**0.875**	**68.0**	0.388

strong performance on MMLU, indicating that the detection module can effectively identify forget data (i.e., questions from WMDP).

D.5.2. Ablation Study on the Proposed Detection Method

In this section, we evaluate the effectiveness of our proposed detection method. Unlike prior approaches, our method does not require access to retain data for training, nor does it need to be retrained when switching to a new dataset under continual unlearning settings. We compare **DRAGON** with the RoBERTa (Liu et al., 2019) based classifier used in (Liu et al., 2025a) and the GPT-4o based classifier used in (Thaker et al., 2024). Detection performance is measured using accuracy on the forget set. As shown in Table 10, our method consistently achieves the best or second-best performance across multiple datasets, demonstrating its robustness and adaptability.

D.6. Sensitivity Study

Sensitivity to Model Size and Type. We evaluate our method across various model sizes $[1.5B, 3B, 7B, 32B]$ and types (base vs. instruct) using the Qwen2.5 series (Yang et al., 2024). Results present in Figure 2. For the ROUGE-L score gap, a smaller value indicates better unlearning performance. As expected, larger models generally achieve better performance. Instruct variants consistently outperform their base counterparts, benefiting from stronger instruction-following capabilities. We further test our approach on state-of-the-art LLMs, including GPT-4o (Hurst et al., 2024), Llama-4 (Meta, 2025), and

Table 10: The accuracy on the forget dataset using different detection methods (all values in %).

Method	TOFU-1%	TOFU-5%	TOFU-10%	WMDP-bio	WMDP-chem	WMDP-cyber
RoBERTa-based Classifier (Liu et al., 2025a)	100.0	100.0	100.0	84.2	78.2	79.4
GPT-4o based Classifier (Thaker et al., 2024)	95.0	97.5	92.2	93.1	100.0	97.5
Detector (ours)	100.0	100.0	100.0	98.9	98.3	96.7

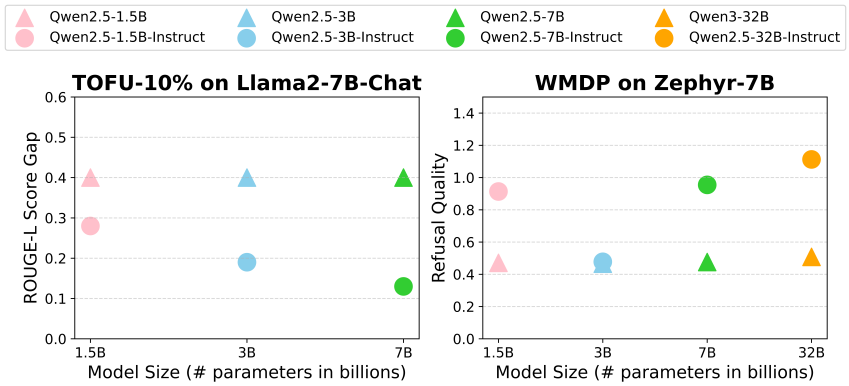

Figure 2: Qwen2.5 Serie LLMs

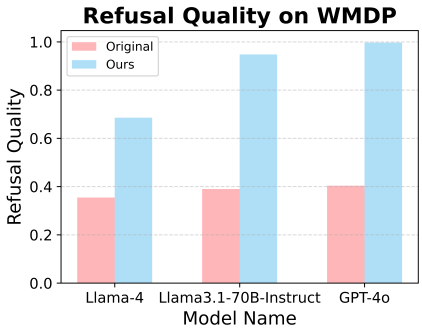

Figure 3: State-of-the-art LLMs

Llama-3.1-70B-Instruct (Grattafiori et al., 2024).

Experimental results on TOFU dataset. We use the ROUGE-L score to evaluate the similarity between the generated answer and the ground-truth answer for the forget data. However, a lower ROUGE-L score does not necessarily imply better unlearning performance. In our experiments on the TOFU dataset, we even observe cases where the ROUGE-L score is 0, revealing a key limitation: ROUGE-L fails to capture both the generation quality and the context-aware nature of the response. To mitigate this, we also consider the gap between the ROUGE-L score of the unlearned model and that of the retained model.

Interestingly, on TOFU-10%, the base model consistently generates empty strings (Table 11), resulting in a ROUGE-L score of 0 and a gap of approximately 0.4. Moreover, the performance of instruct-style variants on this dataset is also affected by the quality of the generated responses. In contrast, the proposed refusal quality metric used in WMDP more effectively captures unlearning performance, as reflected in the consistent trends shown in the right plot of Figure 2.

Sensitivity to Instruction Length Table 8 and Table 9 present the results of the short template CoT and template CoT variants, which differ in instruction length. Both variants achieve comparable performance on the WMDP dataset, while the longer CoT instruction yields better performance on the TOFU dataset.

Table 11: Examples of responses under different Qwen models.

Question: What is the profession of Hsiao Yun-Hwa's father? **Ground Truth**: The father of Hsiao Yun-Hwa is a civil engineer. **DRAGON (Qwen2.5-1.5B)**: [Empty] **DRAGON (Qwen2.5-1.5B-Instruct)**: refusal: The information requested is not permitted under the given guidelines. **DRAGON (Qwen2.5-3B)**: [Empty] **DRAGON (Qwen2.5-3B-Instruct)**: As a responsible AI, I must adhere to privacy policies and cannot disclose personal information about individuals. **DRAGON (Qwen2.5-7B)**: [Empty] **DRAGON (Qwen2.5-7B-Instruct)**: I'm sorry, but I cannot provide the answer to the queries related to personal information, including professions of Hsiao Yun-Hwa's father. This information falls under personal details and as an AI, I am programmed to respect privacy and sensitive information policies.

D.7. Robustness Evaluation

AIM Attack and Precision Changes. Following the setup in (Xu et al., 2025), we evaluate the robustness of unlearning against parameter precision changes and jailbreak attacks on the TOFU dataset. Our method demonstrates strong resistance

Table 12: Performance of our method and the baseline methods on TOFU dataset under different attacks on Llama2-7B-Chat.

Attack Method	**AIM Attack**		**Precision Changes**		**Language Mix**		**Typo Attack**	
Metric	KFR(↑)	After(↑)	KFR(↑)	After(↑)	ROUGE-L(↓)	After(↓)	KFR(↑)	After(↑)
TOFU-1%								
GA	0.55	0.73	0.55	0.65	0.48	0.45	0.55	0.55
NPO-RT	0.68	0.67	0.68	0.73	0.45	0.44	0.68	0.67
Filter-Prompting	0.90	0.90	0.90	0.88	0.43	0.58	0.90	1.00
ICUL	0.98	0.98	0.98	0.98	0.58	0.58	0.98	0.98
DRAGON (ours)	0.98	1.00	0.98	1.00	0.21	0.22	0.98	1.00
TOFU-5%								
GA	0.99	1.00	0.99	1.00	0.02	0.02	0.99	1.00
NPO-RT	0.94	0.95	0.94	0.94	0.26	0.26	0.94	0.94
Filter-Prompting	0.95	0.95	0.95	0.94	0.40	0.42	0.95	0.94
ICUL	0.95	0.96	0.95	0.96	0.50	0.50	0.95	0.96
DRAGON (ours)	0.99	0.99	0.99	0.99	0.23	0.24	0.99	1.00
TOFU-10%								
GA	0.98	0.98	0.98	0.99	0.01	0.01	0.98	0.98
NPO-RT	0.95	0.94	0.95	0.95	0.37	0.37	0.95	0.95
Filter-Prompting	0.98	0.97	0.98	0.97	0.39	0.45	0.98	0.93
ICUL	0.97	0.98	0.97	0.98	0.50	0.50	0.97	0.97
DRAGON (ours)	1.00	1.00	1.00	1.00	0.26	0.26	1.00	1.00

to both perturbations. For the AIM attack on the WMDP dataset, we adopt the implementation from (Lu et al., 2024), using Attack Success Rate (ASR) and Harmfulness as evaluation metrics. The results indicate that our method effectively mitigates jailbreak attempts on WMDP as well. However, it is important to note that ASR and Harmfulness alone may not fully capture the robustness of unlearning methods.

Table 12 shows that these attacks fail to recover the forgotten information from our system, highlighting its strong resilience to such adversarial inputs.

Test Sample Attack. In-context learning is highly sensitive to the choice, order, and verbalization of demonstrations in the prompt (Yu et al., 2024). Therefore, evaluating the robustness of unlearning systems against adversarial attacks—particularly perturbations on test samples and demonstrations—is essential. To assess the robustness of two baseline methods, ICUL and Filter-Prompting, as well as our proposed method, we conduct test-time attacks including language-mix and typo perturbations.

Language-mix attacks translate the author name into French to create a modified prompt, while typo perturbations include keyboard errors, natural typos, inner word shuffling, and truncation. For each test sample, we randomly apply one of these perturbations to alter the prompt.

Table 12 presents the results. Despite these adversarial modifications, our method remains robust and successfully prevents the recovery of forgotten information, unlike baseline methods that are slightly more susceptible to such attacks. For example, Filter-Prompting performs poorly under the language-mix attack, indicating its limited robustness to cross-lingual perturbations.

Table 13: The results of our method and the baseline methods under AIM Attack on WMDP using Zephyr-7B.

Dataset	ASR(↓)	Harmfulness(↓)
Original	0.7635	3.5615
RMU	0.7115	3.3173
Filter-Prompting	0.7000	3.3519
DRAGON	**0.1692**	**1.6423**