Atyaephyra at SemEval-2025 Task 4: Low-Rank Negative Preference Optimization

Jan Bronec and **Jindřich Helcl**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
`janbronec00@gmail.com`, `helcl@ufal.mff.cuni.cz`

Abstract

We present a submission to the SemEval 2025 shared task on unlearning sensitive content from LLMs. Our approach employs negative preference optimization using low-rank adaptation. We show that we can utilize this combination to efficiently compute additional regularization terms, which help with unlearning stabilization. The results of our approach significantly exceed the shared task baselines.

1 Introduction

Transformer-based Large Language Models (LLM) trained on large data corpora have shown remarkable performance in many natural language processing tasks. However, their ability to remember portions of the training data (Carlini et al., 2021) might raise legal, ethical, and other issues. These consist of LLMs regurgitating copyright-protected creative content learned from the Web or private personal information such as social security numbers, addresses, and others. For the latter, regulations such as the EU's General Data Protection Regulation (GDPR) or the California Consumer Privacy Act (CCPA) mandate the right for the removal of such information from the training data as per the "Right to be forgotten".

Unfortunately, these issues are often discovered only after model training. Although discarding such sensitive items from the training data sets and subsequent retraining is possible, it is generally prohibitively expensive. The field of machine unlearning tackles this exact issue by treating the removal of information as model fine-tuning. Although several state-of-the-art approaches and benchmarks exist for LLM unlearning (Zhang et al., 2024a; Maini et al., 2024), the field is still relatively unexplored.

To facilitate further progress in the field, Ramakrishna et al. (2025b) developed a comprehensive evaluation challenge for unlearning sensitive datasets in LLM as a part of the International Workshop on Semantic Evaluation.[1] This paper presents our submission to the shared task.

To solve the task, we utilized the state-of-the-art unlearning method negative preference optimization (NPO; Zhang et al., 2024a). We combined this method with low-rank adaptation (LoRA; Hu et al., 2021) because we consider the computational effectiveness of unlearning to be of high importance. Although several approaches use parameter-efficient methods for unlearning in transformers (Gao et al., 2025; LiM et al., 2024; Ding et al., 2025), the combination of NPO and LoRA is novel.

We show that with LoRA, we can cheaply compute additional regularization terms that use the original model's output distribution without any memory overhead. We further show that including the KL divergence minimization regularization stabilizes the unlearning for a higher number of epochs. Furthermore, our solution significantly outperforms the shared task baselines. We release the source code of our submission on GitHub.[2]

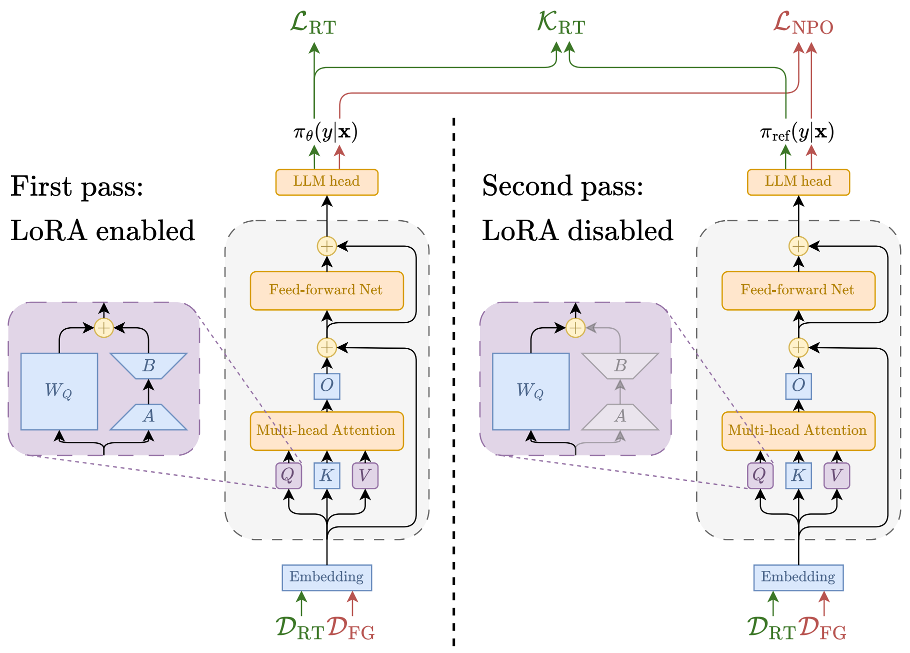
2 Task Background

The goal of the shared task (Ramakrishna et al., 2025b) is to build a method for unlearning information from a given target large language model. For a given model, a forget set $\mathcal{D}_{\text{FG}}$, and a retain set $\mathcal{D}_{\text{RT}}$, the method should be able to remove the information present in the forget set from the model while preserving the data from the retain set and not deteriorating the model's performance on unrelated tasks.

As the target model, the task organizers utilized OLMo (a pre-trained LLM), specifically its 7B and 1B versions (Groeneveld et al., 2024). Since OLMo is trained on an open dataset Dolma (Soldaini et al., 2024), it makes for a good choice for this task.

These target models were further fine-tuned to

[1] `https://llmunlearningsemeval2025.github.io/`
[2] `https://github.com/XelfXendr/peft_unlearning`



$\mathcal{L}_{\text{RT}}$
$\mathcal{K}_{\text{RT}}$
$\mathcal{L}_{\text{NPO}}$
$\pi_\theta(y|\mathbf{x})$
$\pi_{\text{ref}}(y|\mathbf{x})$
First pass:
LoRA enabled
Second pass:
LoRA disabled
LLM head
Feed-forward Net
Multi-head Attention
O
Q K V
W_Q
B
A
Embedding
$\mathcal{D}_{\text{RT}}$ $\mathcal{D}_{\text{FG}}$

Figure 1: Overview of our unlearning method. We augment the model with LoRA and train only the LoRA parameters. For each batch, we compute our training loss shown in Equation 5 in two passes. During the first pass, we leave the LoRA layers enabled and compute the $\mathcal{L}_{\text{RT}}$ term of our loss. During the second pass, we compute the $\mathcal{K}_{\text{RT}}$ and $\mathcal{L}_{\text{NPO}}$ terms by disabling the LoRA layers and only utilizing the backbone. This way, we do not need to maintain a separate copy of the backbone.

remember a dataset consisting of three document types: long-form synthetic creative documents, short-form synthetic biographies containing personally identifiable information such as fake names, phone numbers, or home addresses, and real documents sampled from the Dolma dataset. Each entry in the dataset consists of input-output pairs that cover either sentence completion or question answering.

The organizers split this dataset into separate retain and forget sets, released the training and validation versions of each for the task, and provided an unlearning evaluation framework LUME (Ramakrishna et al., 2025a). The data is in English only.

3 Method Overview

Our approach for this task combines negative preference optimization (NPO; Zhang et al., 2024a) and low-rank adaptation (LoRA; Hu et al., 2021) for parameter-efficient fine-tuning.

3.1 Negative Preference Optimization

Most currently used unlearning approaches are based on gradient ascent (GA). Consider a language model π_θ with parameters θ, which models the next token y distribution based on context x. The basic premise is to ascend the classic next-token prediction cross-entropy loss on the forget data $\mathcal{D}_{\text{FG}}$ instead of descending it:

$$\mathcal{L}_{\text{GA}}(\theta) = \mathbb{E}_{\mathcal{D}_{\text{FG}}}\left[\log \pi_\theta(y|x)\right] \tag{1}$$

Zhang et al. (2024a) showed that the basic gradient ascent quickly deteriorates the utility of the model and proposed NPO as an alternative unlearning strategy. For the original model π_{ref}, and a positive hyper-parameter β, the NPO loss is as follows:

$$\mathcal{L}_{\text{NPO}}(\theta;\beta) = \\ = \mathbb{E}_{\mathcal{D}_{\text{FG}}}\left[\frac{2}{\beta}\log\left(1+\left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)^{\beta}\right)\right] \tag{2}$$

The authors further show that $\nabla_\theta\mathcal{L}_{\text{NPO}}(\theta;\beta)$ converges to $\nabla_\theta\mathcal{L}_{\text{GA}}(\theta)$ as β approaches zero. For positive values of β, the NPO loss effectively dampens the contribution of already unlearned samples.

We further extend $\mathcal{L}_{\text{NPO}}$ with two regularization terms. Zhang et al. (2024a) regularize the NPO loss with a "retain loss" $\mathcal{L}_{\text{RT}}(\theta)$ to improve their unlearning results.

$$\mathcal{L}_{\text{RT}}(\theta) = -\mathbb{E}_{\mathcal{D}_{\text{RT}}}\left[\log \pi_\theta(y|x)\right] \tag{3}$$

As a complement to the unlearning regularization by $\mathcal{L}_{\text{RT}}$, we also utilized the Kullback-Leibler divergence $\mathcal{K}_{\text{RT}}$ between π_θ and π_{ref} on the retained set. This regularization was also used by Maini et al. (2024).

$$\mathcal{K}_{\text{RT}}(\theta) = \mathbb{E}_{\mathcal{D}_{\text{RT}}}\left[KL\left(\pi_\theta(\cdot|x)||\pi_{\text{ref}}(\cdot|x)\right)\right] \quad (4)$$

The final loss $\mathcal{L}(\theta)$ we used for the task is as follows:

$$\mathcal{L}(\theta;\beta,\gamma,\delta) = \\ = \mathcal{L}_{\text{NPO}}(\theta;\beta) + \gamma\mathcal{L}_{\text{RT}}(\theta) + \delta\mathcal{K}_{\text{RT}}(\theta) \quad (5)$$

A significant issue with this approach is that $\mathcal{L}_{\text{NPO}}$ and $\mathcal{K}_{\text{RT}}$ both require us to maintain the output log-probs of the original π_{ref}. In the case of $\mathcal{L}_{\text{NPO}}$, we only need the log-prob of the specific token in the training set, which we can pre-compute before unlearning. Unfortunately, we need the entire output token distribution for $\mathcal{K}_{\text{RT}}$, which is unfeasible to precompute and save for each token in a sufficiently large training set. Therefore, we must compute it on demand, for which we would typically need to keep a copy of the original model in memory.

Our contribution comes from the insight that with LoRA, we can circumvent the need to keep a copy of the original model because the original model weights are still present within the augmented model. When we need to compute $\pi_{\text{ref}}(\cdot|x)$ for $\mathcal{K}_{\text{RT}}$, we ignore the LoRA transformations. We show the overall workflow of our method in Figure 1. In addition to that, LoRA substantially reduces the memory requirements for fine-tuning using AdamW (Loshchilov and Hutter, 2019).

3.2 Low-Rank Adaptation

The individual attention layers of OLMo each contain four linear transformations W_q, W_k, W_v, $W_o \in \mathbb{R}^{d\times d}$. The width and height d of the matrices are 2048 for OLMo-1B and 4096 for OLMo-7B. The AdamW optimizer stores the gradient moment estimations of these matrices, which poses a significant memory cost for model fine-tuning.

The idea of LoRA is to enhance some of the linear transformations $y := Wx$ within the attention layers of an LLM with a decomposed low-rank linear transformation. For matrices $A \in \mathbb{R}^{r\times d}, B \in \mathbb{R}^{d\times r}$ the augmented transformation becomes $y := Wx + \frac{\alpha}{r}BAx$. The value r is the rank of the transformation, and α is a hyper-parameter constant in r. The factor $\frac{\alpha}{r}$ is often introduced to counteract the effect that increasing r has on the effective learning rate. The original weights W are then frozen during fine-tuning and only the matrices A, B are updated. This drastically decreases the number of trained parameters as r can generally be set to a fairly low value, such as 2 or 5. The memory requirements of AdamW are thus significantly reduced as well.

LoRA has a further benefit over other parameter-efficient fine-tuning methods, such as adapters (Houlsby et al., 2019) and Quantized Side Tuning (Zhang et al., 2024b) in that the we can merge the low-rank matrices into the original weight matrix $W' := W + \frac{\alpha}{r}BA$ after we finish fine-tuning. This allows us to update the model without affecting its architecture.

4 Experiments

For our experiments, we focused on the fine-tuned OLMo-7B model, which we unlearned using the training retain and forget sets, all provided by the task organizers. We chose a fixed value $\beta = 0.5$ of the NPO loss hyper-parameter and $r = \alpha = 5$ for LoRA, as these values gave us reasonably good results. We experimented with various values for the hyper-parameters γ and δ. The learning rate was set to 10^{-4} with a batch size of 4. We perform a broader hyper-parameter search on the OLMo-1B model in Appendix A.

We used four combinations of values for γ, δ. In three of them, we set $\delta = 1$ and experimented with values $0, 0.5$, and 1 for γ to determine the effect of $\mathcal{K}_{\text{RT}}$ on the deterioration of the model. In the last run, we only kept $\mathcal{K}_{\text{RT}}$ with $\delta = 1$ and set $\gamma = 0$. We ran five independent runs with unique seeds for each of the combinations.

Following the shared evaluation scheme, we measure four scores to evaluate the quality of our solution. First, we calculate the ROUGE-L score (Lin, 2004) for each sample in the validation sets. By computing the score separately for sentence-completion and question-answering pairs of each document type in the forget and retain sets, we obtain 12 values. We invert the values for the forget sets and produce the Task score by merging the 12 resulting values using the harmonic mean.

We perform a loss-based Membership Inference Attack (MIA; Duan et al., 2024) and scale the resulting MIA score S_{MIA} to penalize both under-

Run	Epoch	Task score ↑		MIA score ↑		MMLU ↑		Final score ↑	
		μ	σ	μ	σ	μ	σ	μ	σ
$\gamma = 1, \delta = 0.0$	10	.431	.020	.657	.269	.461	.010	**.516**	.089
	20	.449	.029	.389	.117	.449	.008	.429	.043
$\gamma = 1, \delta = 0.5$	10	.349	.045	.375	.193	.462	.005	.395	.069
	20	.434	.021	.594	.201	.439	.017	.489	.074
$\gamma = 1, \delta = 1.0$	10	.349	.041	.165	.083	**.465**	.005	.327	.027
	20	**.453**	.016	.620	.219	.449	.016	.507	.072
$\gamma = 0, \delta = 1.0$	10	.369	.080	**.699**	.170	.441	.020	.503	.056
	20	.332	.085	.400	.113	.370	.054	.367	.021
Baseline NPO	–	.021	–	.080	–	.463	–	.188	–
Baseline GD	–	0	–	.382	–	.348	–	.243	–
Original model	*0*	*0*	–	*0*	–	*.510*	–	*.170*	–

Table 1: Unlearning results on fine-tuned OLMo-7B-0724-Instruct-hf for various hyper-parameters. The first four entries correspond to our experimental runs with different hyper-parameter choices. Each run was repeated five times with different seeds. We report the means and standard deviations of the scores after the 10th and 20th epochs, estimated from those five runs. We compare our results to the baselines reported by task organizers. "GD" stands for Gradient Difference (Liu et al., 2022).

and over-training: $S'_{\text{MIA}} := 1 - 2 \cdot |S_{\text{MIA}} - 0.5|$. To measure the degradation of the model after unlearning, we use the MMLU benchmark (Hendrycks et al., 2021). Finally, we compute the Final score as the arithmetic mean of the previous three scores.

5 Results

In Table 1, we report the mean values and standard deviations of the different scores estimated over the five unlearning runs. We evaluated the scores using the provided validation sets. We show the results after 10 and 20 unlearning epochs. We compare our results with the baselines provided by the shared task organizers.[3] In Figure 2, we show the development of the different scores throughout the epochs.

In general, our runs considerably outperform the provided baseline solutions. The runs that used $\mathcal{L}_{\text{RT}}$ overall did not degrade the MMLU score as much as those where only $\mathcal{K}_{\text{RT}}$ was involved. Including $\mathcal{K}_{\text{RT}}$ with positive δ in the loss slowed the training. However, since achieving a high MIA score required hitting a sweet spot between under- and over-training, including $\mathcal{K}_{\text{RT}}$ helped stability in the long run.

We placed lower than expected on the 7B model in the task leaderboard. This was caused by a logical error in our submission script, which effectively caused our solution to perform only three unlearning epochs instead of 20. We fixed our script for the 1B model evaluation, on which we ranked as expected.

[3] https://llmunlearningsemeval2025.github.io/

6 Conclusion

The combination of NPO and LoRA proved to be an effective strategy for LLM unlearning. In addition to significant memory usage improvements, LoRA allows us to cheaply compute an additional regularization term, stabilizing the unlearning for a higher number of epochs. In our future work, we wish to further investigate the effect of LoRA and its rank on the resilience of the model to quality deterioration.

Acknowledgments

We would like to thank Jindřich Libovický for his comments on the draft of the paper.

This work was supported by the project "Human-centred AI for a Sustainable and Adaptive Society" (reg. no.: CZ.02.01.01/00/23_025/0008691), co-funded by the European Union, and by the Charles University project PRIMUS/23/SCI/023.

Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

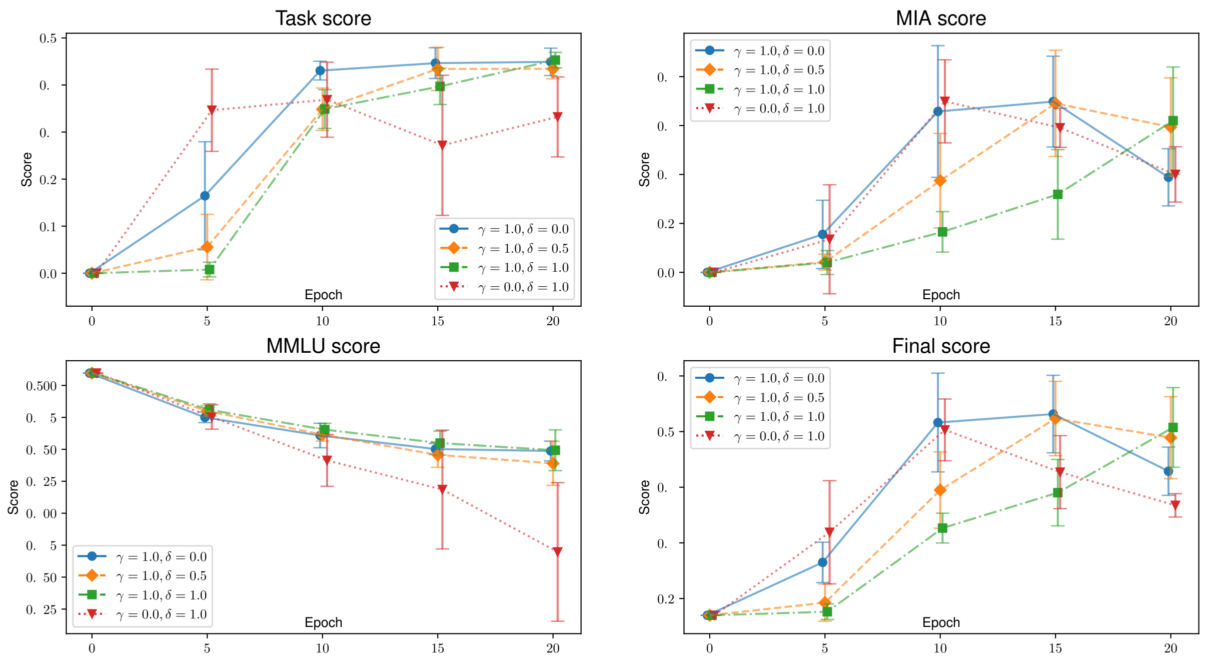


Figure 2: Unlearning results on fine-tuned OLMo-7B-0724-Instruct-hf for various hyper-parameters. Measured scores are averaged over five randomly seeded runs. The standard deviation estimates are shown in error bars. (points are offset for better visibility)

References

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

CCPA. An act to add title 1.81.5 (commencing with section 1798.100) to part 4 of division 3 of the civil code, relating to privacy. Accessed: Feb. 24, 2025.

Chenlu Ding, Jiancan Wu, Yancheng Yuan, Jinda Lu, Kai Zhang, Alex Su, Xiang Wang, and Xiangnan He. 2025. Unified parameter-efficient unlearning for LLMs. In *The Thirteenth International Conference on Learning Representations*.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? In *First Conference on Language Modeling*.

Chongyang Gao, Lixu Wang, Kaize Ding, Chenkai Weng, Xiao Wang, and Qi Zhu. 2025. On large language model continual unlearning. In *The Thirteenth International Conference on Learning Representations*.

GDPR. Do we always have to delete personal data if a person asks? Accessed: Feb. 24, 2025.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and

Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Guihong LiM, Hsiang Hsu, Chun-Fu Richard Chen, and Radu Marculescu. 2024. Fast-ntk: Parameter-efficient unlearning for large-scale models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 227–234.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Bo Liu, Qiang Liu, and Peter Stone. 2022. Continual learning and private unlearning. In *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pages 243–254. PMLR.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. Tofu: A task of fictitious unlearning for llms. *Preprint*, arXiv:2401.06121.

Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. 2025a. Lume: Llm unlearning with multitask evaluations. *Preprint*, arXiv:2502.15097.

Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. 2025b. Semeval-2025 task 4: Unlearning sensitive content from large language models. *Preprint*, arXiv:2504.02883.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024a. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*.

Zhengxin Zhang, Dan Zhao, Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Qing Li, Yong Jiang, and Zhihao Jia. 2024b. Quantized side tuning: Fast and memory-efficient tuning of quantized large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–17, Bangkok, Thailand. Association for Computational Linguistics.

A Additional experiments

To choose the parameters for our task submission, we performed two hyper-parameter searches on the provided OLMo-1B model. For the first search, we initially set $r = \alpha = 5$ for the LoRA parameters and searched for optimal values of parameters γ, δ from our training loss shown in Equation 5. We experimented with utilizing only one of the regularization terms by setting one of the γ, δ parameters to zero, as well as combining the two terms with various factors. We kept the NPO regularization parameter constant with the value $\beta = 0.5$. We used the Adam optimizer with a learning rate of 10^{-4} and a batch size of 4 sequences. The fine-tuning ran for 20 epochs. We repeated each run five times with different seeds and averaged the resulting scores. We show the results in Table 2.

Overall, increasing γ and δ led to an increase in the task score. However, too high values of γ or δ led to a drop in the MIA score. On average, the combination of $\gamma = 1, \delta = 0.5$ gave us the best final score. We used this combination for our task submission and for further experiments on the OLMo-7B model, which we discussed in Section 4.

With the parameters $\gamma = 1, \delta = 0.5$, we also performed a second hyper-parameter search for the optimal value of the LoRA rank r. For this search, we kept the value of $\alpha = 5$ constant, as suggested by Hu et al. (2021). We kept β, the learning rate, batch size, and the number of epochs the same as in the previous search. Once again, we ran five experiments for each value of r, and averaged the resulting scores. The results can be seen in Table 3. In Figure 3, we show the development of the scores throughout the epochs. Although some of the runs exhibit better average performance in the various scores, we find the variance in our results too high to draw conclusions. Ultimately, we settled with the rank $r = 5$ for our submission, as we did not see any benefits in increasing the number of parameters with higher ranks.

Hyper-params		Task score ↑		MIA score ↑		MMLU ↑		Final score ↑	
γ	δ	μ	σ	μ	σ	μ	σ	μ	σ
0.5	0.0	.387	.033	.322	.047	.258	.008	.323	.022
1.0	0.0	.398	.041	.540	.054	.265	.005	.401	.029
2.0	0.0	.423	.056	.317	.167	**.270**	.003	.337	.049
0.0	0.5	.269	.086	.285	.033	.256	.006	.270	.035
0.0	1.0	.245	.066	.516	.074	.262	.006	.341	.040
0.0	2.0	.299	.059	.599	.141	.269	.004	.389	.050
0.0	5.0	.362	.065	.039	.049	.267	.007	.222	.021
0.2	0.2	.379	.030	.295	.056	.256	.011	.310	.019
0.5	0.5	.350	.066	.464	.104	.268	.006	.361	.048
0.5	1.0	.442	.030	.719	.056	.261	.009	.474	.025
1.0	0.5	.394	.036	**.821**	.117	.266	.006	**.494**	.038
1.0	1.0	.419	.061	.721	.198	.270	.004	.470	.084
1.0	2.0	.400	.081	.606	.202	.265	.003	.424	.071
2.0	1.0	**.469**	.027	.487	.201	.269	.004	.408	.062

Table 2: Results of a hyper-parameter search for regularization parameters γ and δ conducted on the fine-tuned OLMo-1B model. Each run was repeated five times with different seeds, and we report the mean and standard deviation estimates of the scores after the 20th epoch. The scores were evaluated on validation sets provided for the shared task.

Hyper-params		Task score ↑		MIA score ↑		MMLU ↑		Final score ↑	
r	α	μ	σ	μ	σ	μ	σ	μ	σ
1	5	.435	.053	.743	.107	.266	.005	.481	.047
2	5	**.455**	.033	**.895**	.127	.266	.005	**.539**	.039
5	5	.417	.059	.765	.097	**.272**	.005	.485	.041
10	5	.372	.037	.835	.058	.269	.008	.492	.025
25	5	.417	.030	.713	.227	.265	.006	.465	.075
100	5	.432	.042	.658	.282	.271	.006	.454	.096

Table 3: Results of a hyper-parameter search for LoRA ranks r conducted on the fine-tuned OLMo-1B model. The regularization parameters were set to $\gamma = 1, \delta = 0.5$. Each run was repeated five times with different seeds and we report the mean and standard deviation estimates of the scores after the 20th epoch. The scores were evaluated on validation sets provided for the shared task.

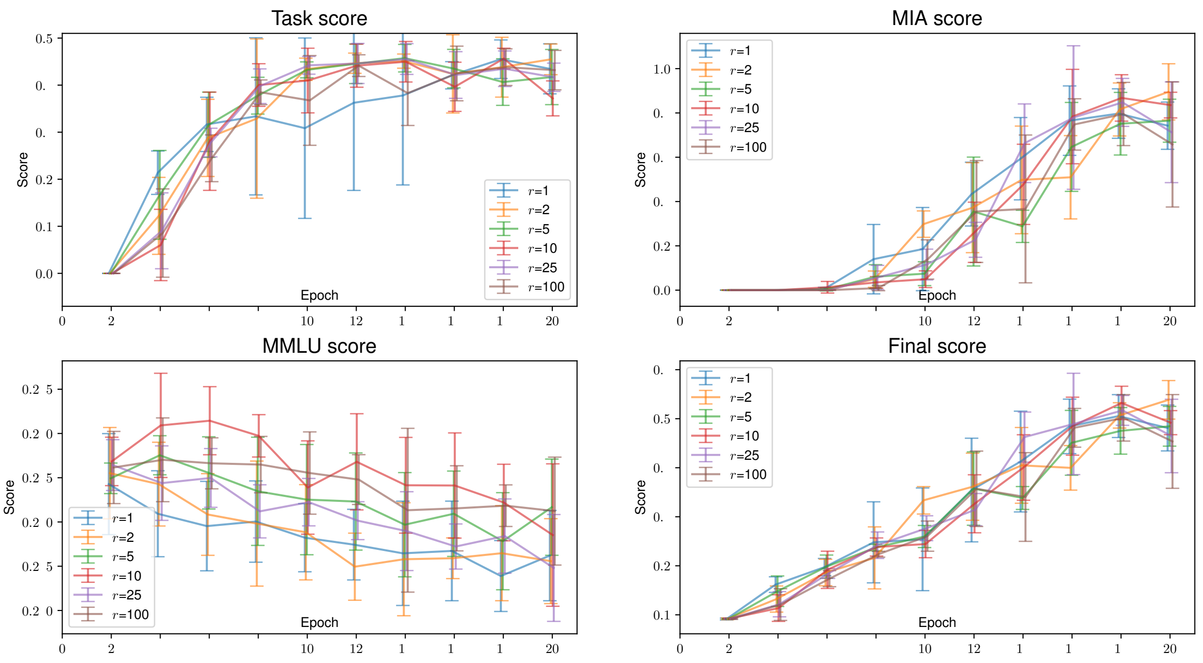


Figure 3: Unlearning results on fine-tuned OLMo-1B for various values of LoRA rank r. Measured scores are averaged over five randomly seeded runs. The standard deviation estimates are shown in error bars. (Points are offset for better visibility.)