

Recover-LoRA: Data-Free Accuracy Recovery of Degraded Language Models via Low-Rank Adaptation

Devleena Das, Rajeev Patwari, Ashish Sirasao

Advanced Micro Devices, Inc. (AMD)

{devleena.das, rajeev.patwari, ashish.sirasao}@amd.com

Abstract

Inference optimizations such as quantization, pruning, format and datatype conversion, model export, and serialization can lead to functional degradations in language model task performance. While most efforts on performance recovery for deployment focus on robust quantization techniques, we focus on recovering model accuracies from any sources that degrade model weights, such as improper model serialization. In this work, we propose Recover-LoRA, a lightweight and dataset agnostic method to recover accuracy in degraded models. Recover-LoRA uses synthetic data and logit distillation to learn LoRA adapters on selective layers that facilitate aligning the degraded model to its full precision model. We investigate the utility of Recover-LoRA across a diverse set of small language models (SLMs), including models with varying attention architectures, multi-head attention (MHA) and group-query attention (GQA), as well as several evaluation datasets. Our results show that Recover-LoRA recovers model accuracies by 5-17% on MHA and GQA SLMs.

1 Introduction

Small language models (SLMs), typically under 5B parameters, have shown strong capabilities on downstream tasks while offering a smaller memory and compute footprint compared to their larger language model counter parts (i.e. Phi3.5-mini, Llama3.2 1B, etc.)(Lu et al., 2024). These smaller models have become of popular interest for edge deployment where compute, memory and latency are critical bottlenecks¹. However, for edge deployment, language models often undergo further optimization or conversion steps that can inadvertently introduce accuracy degradation due to structural inconsistencies or weight corruptions. For

example, accuracy loss can stem from quantization (Zhu et al., 2024), sparsity (Zafrir et al., 2021), improperly saving or loading model states, custom layers, or format drifts when transferring among tool chains (eg. Pytorch to ONNX). These scenarios may result in packaged models that are structurally sound, where shapes and architectures are preserved, but downstream task performance significantly varies from the original model.

Among these sources of error, quantization is one of the most popular studied, given its importance for reducing inference latency and memory footprint (Zhu et al., 2024). Post-Training Quantization (PTQ) methods such as AWQ (Lin et al., 2024) convert weights to lower precision without retraining, while Quantize-Aware Training (QAT) retrains the model with simulated quantization noise (Lang et al., 2024). Most recently, LLM-QAT (Liu et al., 2023) has shown that synthetic data, instead of labelled data, can be used to perform QAT for LLama models.

While our work is inspired by QAT and LLM QAT to preserve model accuracy, we focus on recovering accuracy from more sources of errors, outside of quantization, that can occur in deployment settings, leading to corrupted model weights. Additionally, we consider the practical constraints within industry settings such as scarce labeled data or proprietary data, and minimizing retraining of large models. To this end, we explore the following: *how can we recover model accuracy loss without requiring full, model training and utilize synthetic data?*

In this work, we introduce **Recover-LoRA** a lightweight, dataset agnostic approach to recovering accuracy from functionally degraded models where the model weights have undergone silent corruption. Recover-LoRA leverages synthetic data, inspired by LLM QAT (Liu et al., 2023), to learn low-rank matrices (LoRA adapters (Shen et al.)) that align the corrupted model with its full-

¹<https://blogs.windows.com/windowsexperience/2024/12/06/phi-silica-small-but-mighty-on-device-slm/>

precision, reference language model via logit distillation (Gou et al., 2021). In this manner, Recover-LoRA provides a parameter-efficient approach to accuracy recovery, while providing data independence through synthetic data. While LoRA (Shen et al.) is a common lightweight finetuning approach, it is traditionally applied to task adaptation with labeled datasets. To the best of our knowledge, Recover-LoRA is the first to consider the feasibility of LoRA adapters in recovering degraded model accuracy.

We study the efficacy of Recover-LoRA across four different SLM architectures, including multi-head and group-query attention models (MHA, GQA), using functionally degraded models derived from improper model weight serialization, and evaluate on seven different datasets. Our work contributes the following:

1. We introduce Recover-LoRA, to the best of our knowledge, as the first approach to recover lost model accuracy in degraded models. Recover-LoRA provides a lightweight and data-flexible method to restore model performance by learning LoRA adapters with logit distillation, and using synthetic data.
2. We demonstrate that Recover-LoRA effectively improves model accuracy in MHA and GQA style models. We show an average accuracy recovery ranging from 5% to 17%, surpassing the recovery capabilities of LLM QAT (Liu et al., 2023) across all tested models, and surpassing dataset-specific LoRA finetuning on three out of the four tested models.

2 Related Work

2.1 Sources of LLM Accuracy Degradation

LLM accuracy degradation can occur due to several factors including quantization (Zhu et al., 2024), sparsity (Zafrir et al., 2021), framework conversion (Louloudakis et al., 2023), datatype conversion (Rouhani et al., 2023), etc. Below we describe key sources of error related to our work.

Recently, Jalal et al. (Jajal et al., 2023) highlight the common failure points in ONNX conversion, whereas FetaFix (Louloudakis et al., 2023) proposes an automated approach to detect and repair models conversions between deep learning frameworks. Similarly, state of the art accelerators support fast microscaling formats for inference (Rouhani et al., 2023) like *MXFP6*, *MXFP8*,

and *MXINT8*. Post-training model conversion to such data types may degrade the quality of the LLM specific to the application. Additionally, sparsity techniques that aim to prune model weights can also lead to degraded model performance (Zafrir et al., 2021). These scenarios indicate that conversions for deployment can lead to degraded model performance, highlighting a need for accuracy recovery. Recover-LoRA aims to provide a lightweight method for recovering degraded model performance specifically considering silent failures from model weight serialization.

Quantization for LLMs Quantize-Aware Training (QAT) techniques are widely adopted to reduce impact of quantization specific accuracy degradation. For example, DL-QAT combines group-wise scaling with LoRA based updates to further improve QAT efficiency (Ke et al., 2025). While most QAT approaches use labeled data, LLM QAT (Liu et al., 2023) shows the utility of synthetic data for QAT. LLM QAT (Liu et al., 2023) generates synthetic training data from a full-precision LLaMA 7B model and uses knowledge distillation to train several quantized LLaMA models.

Our work is inspired by the usage of synthetic data in LLM QAT (Liu et al., 2023), but we focus on error stemming from functional degradation not limited to quantization. Specifically, we use synthetically generated data from a pretrained SLM to align the degraded model. Also, unlike LLM QAT, we limit model updates to solely LoRA adapters and enable a more efficient method to accuracy recovery in degraded models.

2.2 Pruning and Recovery Techniques

Recent work has also explored compressing LLMs via pruning and recovering performance post-compression. For example, Minitron (Sreenivas et al., 2024) introduces a multi-stage pipeline involving teacher correction using labeled datasets, followed by structured pruning and knowledge distillation to produce competitive, optimized models. Additionally, Thangarasa et al. (Thangarasa et al., 2024) propose a self-data distillation approach to recover accuracy in pruned models. Specifically, the authors utilize existing fine-tuning datasets and access to a full teacher model to generate distilled outputs which are then used for accuracy recovery. Both Thangarasa et al. (Thangarasa et al., 2024) and Minitron (Sreenivas et al., 2024) rely on access to labelled datasets, whereas our Recover-LoRA

operates in a data-free setting without any reliance on labelled data.

2.3 Parameter-Efficient Fine Tuning (PEFT)

PEFT updates a smaller set of model parameters, compared to all model parameters, to improve computational efficiency during the training process (Ding et al., 2023). A common PEFT approach is LoRA (Shen et al.) in which low-rank matrices, known as adapters, are learned during finetuning. In some PEFT methods the pretrained model is quantized while the LoRA adapters are trained in higher precision. For example, in QLoRA (Dettmers et al., 2023) the pretrained model is quantized to NF4, whereas in QA-LoRA (Xu et al., 2023), it is quantized to INT4. LoRA is primarily motivated to improve training efficiency for task-specific adaptation (Mao et al., 2025). Our work studies the use of LoRA beyond task adaptation and considers the utility of LoRA as a lightweight approach to recover functionally degraded model accuracy due to improper serialization.

3 Background

3.1 LoRA

LoRA (Shen et al.) is a PEFT approach in which low-rank matrices, known as adapters, are trained and added to the pretrained model’s frozen weights. Let $W \in R^{d \times k}$ represent the pretrained weights where d and k define the output and input dimensions. LoRA then defines two trainable matrices $A \in R^{r \times k}$ and $B \in R^{d \times r}$ where $r \ll (d, k)$ represents the rank of the LoRA matrices. During finetuning, W is frozen and only A and B are updated. The LoRA output, Y , for a given layer is then represented as:

$$Y = WX + \alpha BAX \quad (1)$$

where X represents the input activation, and α represents a scaling factor that controls the contribution of LoRA adapters on Y .

3.2 Knowledge Distillation

Knowledge distillation aligns the outputs of smaller student model with the outputs of a larger teacher model (Hinton et al., 2015; Gou et al., 2021; Liu et al., 2023). Let M_T represent the teacher model and M_S represent the student model. During training, M_S is optimized by minimizing the Kullback-Leibler (KL) divergence between the predicted probabilities of M_S , p_s , and the soft-target probabilities of M_T , p_t (Sanh et al., 2019). The loss function is defined as:

Model	L2 Norm
AMD-Olmo-SFT 1B	44.06
Llama3.2 1B	52.97
Gemma2 2B	35.94
DeepSeekR1 Distill Qwen 1.5B	40.69

Table 1: L2 norm difference between original and perturbed weights, indicating model degradation.

$$L_{KD} = KL(p_t || p_s) = \sum_i p_t^i \log \frac{p_t^i}{p_s^i} \quad (2)$$

3.3 Functionally Degraded Models

Pretrained model accuracy degradation can be caused by many factors such as improper serialization, quantization, sparsity, and ONNX export, to name a few. We simulate improper weight serialization by introducing minor perturbations to the attributes of `torch.nn.Linear` for K and V projection layers and save the pretrained model using the HuggingFace `save_pretrained()` API. The result is noisy saved model weights that deviate from the original weights.

4 Methodology

Figure 1 provides an overview of our approach, Recover-LoRA, which aims to recover accuracy lost in functionally degraded models in a lightweight and dataset agnostic manner. Specifically, Recover-LoRA takes as input, M_S , the degraded model, and M_T , the pretrained, full precision model. Note, Recover-LoRA does not require any knowledge of the type of functional degradation, and instead only requires access to M_S and M_T . In our application, we assume M_S has degraded performance due to improper weight serialization. Recover-LoRA then learns key LoRA adapters to align the adapter weights to the pretrained language model’s weights via logit distillation. Training only LoRA adapters makes Recover-LoRA a lightweight approach to recover error from weight-corrupted models. Additionally, the dataset D_{syn} utilized for training is not a labeled dataset. Instead, D_{syn} represents synthetic data generated using the hybrid sampling method outlined in LLM QAT (Liu et al., 2023), making the Recover-LoRA training process data-flexible.

4.1 Functionally Degraded LLM

We insert error into the LLM by introducing minor perturbations to the weight attributes of `torch.nn.Linear` for K and V projections and saving the model with HuggingFace’s `save_pretrained()`.

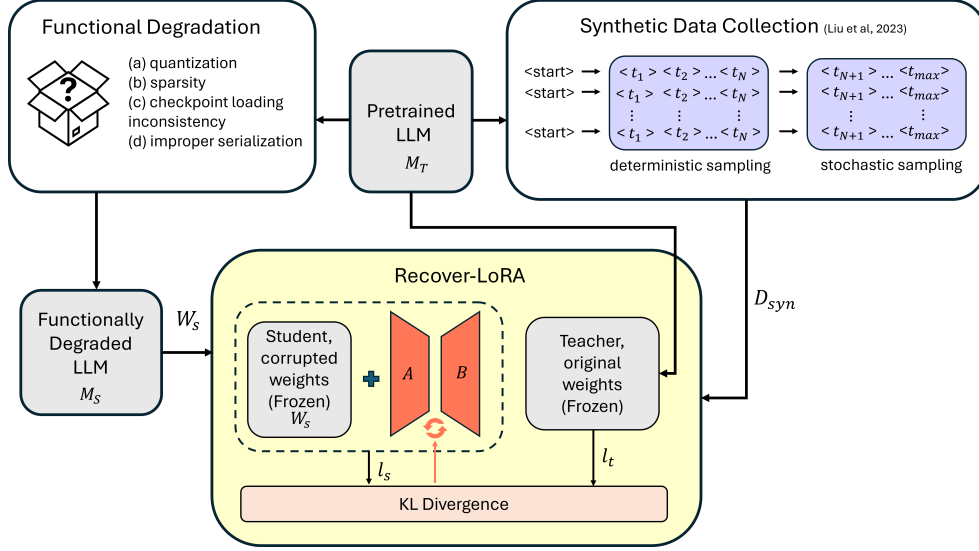


Figure 1: Recover-LoRA recovers model accuracy by leveraging logit distillation to align an improper weight serialized model, M_S , to its pretrained LLM, M_T , by learning LoRA adapters, A and B , with a synthetically generated dataset D_{syn} .

Our functionally degraded LLM simulates incorrect weight serialization. In Table 1, we show the L2 norm difference between the original weights and perturbed weights for the first K projection layer of several models to indicate the random noise that is introduced.

4.2 Synthetic Data Collection

The LoRA adapters in Recover-LoRA are trained with synthetic data generated through a hybrid sampling strategy outlined in LLM QAT (Liu et al., 2023). Specifically, a pretrained language model deterministically generates the first 3-5 tokens, and stochastically generates the remaining tokens, balancing stability and diversity. While LLM QAT (Liu et al., 2023) studies hybrid sampling in a QAT setting, we explore the utility of synthetic data in broader functionally degraded model settings. Details on the hybrid sampling hyperparameters used in Recover-LoRA are provided in Appendix A.

4.3 Recover-LoRA

Recover-LoRA aims to improve the accuracy of M_S by learning a set of lightweight LoRA adapters, A and B , using logit distillation. From Equation 1, the weight matrix W in Recover-LoRA is represented as W_s , the frozen, corrupted weight matrix from improper weight serialization (see Sec. 3). Adapters A and B are optimized by minimizing the KL divergence between the predicted logit distributions of M_S and M_T . Following Equation 2, p_s and p_t represent student and teacher logits, l_s and l_t .

The logits are derived as $l_t = \text{softmax}(M_T(x))$ and $l_s = \text{softmax}(M_S(x))$, where x is a training sample.

While LoRA is a PEFT method for task adaptation, we examine a new use case of LoRA adapters, focusing on restoring model accuracy in degraded models due to corrupted weights. In Section 6, we demonstrate the success of Recover-LoRA in recovering degraded model accuracies in both a parameter and a data-efficient manner.

5 Experiments

We detail the experimental setup used to evaluate Recover-LoRA. We finetune using AMD MI300X GPUs and describe all hyperparameters in Appendix B.

5.1 Baselines

LLM QAT* Our primary baseline is LLM QAT (Liu et al., 2023), which uses synthetic data generated for QAT, via knowledge distillation, to produce quantized LLaMA models. We compare with an adaptation of LLM QAT, **LLM QAT***, where we do not perform the original QAT process of simulating quantization effects in training. Instead, LLM QAT takes an improper serialized model and performs logit distillation on all the model parameters to align the degraded model to its pretrained, teacher model using synthetic data.

SFT LoRA We also compare with the traditional supervised finetuning (SFT) LoRA approach which

	Method	LoRA Adapters	HellaSwag	MMLU Avg.	Arc C	WinoGrande	PiQA	OpenbookQA	BoolQ	Avg	AR%
AMD OLMO SFT 1B	M_T	–	28.38	33.42	24.32	51.3	61.43	18.2	55.44	38.93	–
	M_S	–	25.42	31.41	21.84	50.28	53.05	15.4	43.85	34.46	–
	Recover-LoRA	K,V	25.54	17.96	20.56	50.83	53.97	15.6	62.17	35.23	17.24
	LLM-QAT*	–	27.69	27.71	21.08	50.83	55.88	15.2	39.63	34.00	-10.34
	SFT LORA	K,V	24.48	31.84	23.29	48.46	52.94	18.8	46.21	35.15	15.29
LLAMA3.2 1B	M_T	–	47.74	41.77	31.48	60.93	74.27	26.8	63.73	49.53	–
	M_S	–	25.51	28.58	21.93	50.83	54.3	17.00	37.89	33.72	–
	Recover-LoRA	ATTN, MLP	25.69	32.06	21.33	50.12	53.54	16.8	51.31	35.84	13.38
	LLM-QAT*	–	25.72	17.96	20.73	48.78	53.75	14.6	38.17	31.39	-14.75
	SFT LORA	ATTN, MLP	25.59	23.66	22.01	49.41	51.85	18.6	51.99	34.73	6.39
GEMMA2 2B	M_T	–	54.99	56.75	46.84	68.75	78.67	31.4	73.58	58.71	–
	M_S	–	25.92	22.81	20.73	50.9	53.05	17.6	45.93	33.85	–
	Recover-LoRA	ATTN, MLP	25.98	17.76	20.73	50.51	52.72	14.6	41.68	31.99	-7.45
	LLM-QAT*	–	26.26	24.61	18.26	48.38	54.9	11.4	28/13	31.71	-8.62
	SFT LORA	K,V	35.21	25.04	24.23	52.09	67.37	21.00	61.22	40.88	28.28
DeepSeek R1 Distill Qwen 1.5B	M_T	–	36.39	44.9	34.47	55.88	65.29	20.2	68.01	46.45	–
	M_S	–	25.93	20.64	21.08	48.54	52.83	16.4	59.6	35.00	–
	Recover-LoRA	K,V	26.52	22.85	18.52	49.72	54.79	15.2	61.38	35.6	4.95
	LLM QAT*	–	26.04	19.05	20.31	50.36	54.35	14.00	59.72	34.93	-1.49
	SFT LORA	K,V	27.75	27.29	20.73	49.8	57.24	14.4	48.44	35.09	0.79

Table 2: Average accuracy recovery percentage (AR%) comparisons for all recovery techniques and model comparisons. Note, M_T represents the pretrained SLM, and M_S is the degraded model.

uses good quality labeled datasets to finetune the degraded model, via a cross-entropy loss. Specifically, we leverage the OpenHeremes-2.5, WebInstructSub and Code-Feedback datasets for finetuning, which prior work² has established appropriate for seeing improvements on our designated evaluation tasks. By using these labeled datasets, we measure the effect of using synthetic data and logit distillation compared to high-quality labeled data for model accuracy recovery.

5.2 Evaluation Datasets and Models

Evaluation Datasets We evaluate across seven different datasets. Specifically, we evaluate commonsense reasoning with PiQA (Bisk et al., 2020), OpenBookQA (Mihaylov et al., 2018), WinoGrande (Sakaguchi et al., 2021), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), Arc Challenge (ARC C) (Clark et al., 2018) and multi-task factual knowledge with three randomly selected subsets of MMLU (Hendrycks et al., 2020), MMLU Philosophy, Management, Astronomy.

Evaluation Models We apply Recover-LoRA to four SLMs to assess its generalizability in recovering degraded model accuracy: **Gemma2 2B** (Team et al., 2024), **Llama3.2 1B** (Grattafiori et al., 2024), **DeepSeek-R1-Distill-Qwen 1.5B** (Guo et al., 2025) and **AMD-Olmo-SFT 1B**³. These models represent a diverse set of architectures with different attention mechanisms (see Appendix C for more details).

5.3 Metrics

We define **Accuracy Recovery Percentage (AR%)**, to measure the efficacy of Recover-LoRA:

$$AR\% = \frac{(E_S^* - E_S)}{|E_S - E_T|} * 100 \quad (3)$$

where, E_S and E_T represent evaluation scores of the degraded (M_S) and full-precision (M_T) SLMs, respectively. Additionally, E_S^* represents the evaluation scores of the functionally degraded model, after applying one of the three error recovery techniques, Recover-LoRA, LLM QAT* or SFT LORA. The metric computes how much accuracy is recovered from the degraded model, via a given recovery technique. If $AR\% = 100$, all error is recovered

² <https://huggingface.co/amd/AMD-OLMo-1B-SFT>

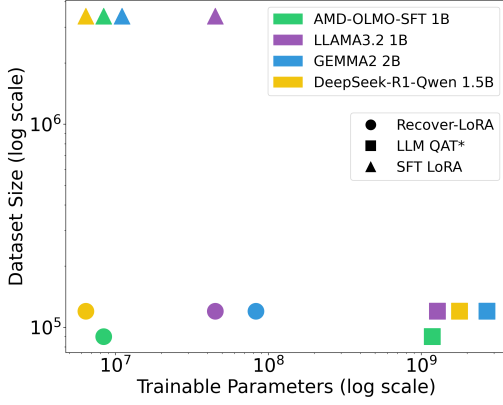


Figure 2: Trainable parameters and dataset size comparisons for all recovery methods, showing the parameter and data efficiency of Recover-LoRA.

and $E_S = E_T$, and if $AR\% = 0$, no error is recovered. If $AR\% < 0$, the recovery technique worsened the error.

6 Results

6.1 Accuracy Recovery from Recover-LoRA

Table 2 shows the average $AR\%$ of each recovery method. Overall, we observe that Recover-LoRA outperforms LLM QAT* and SFT LoRA across three of the four models: AMD-OLMO-SFT 1B, LLaMA3.2 1B and DeepSeek-R1-Distill-Qwen 1.5B. Interestingly, we see negative $AR\%$ from LLM QAT* in all models, showing that LLM QAT* worsens the functional degradation error. We hypothesize that this is due to LLM QAT* updating all model parameters, which may cause overfitting, whereas Recover-LoRA updates a smaller fraction of model parameters. Also, we observe that SFT LoRA performs best for GEMMA2 2B, suggesting that training with synthetic data may be ineffective in some models. We hypothesize that GEMMA2 2B’s architecture may be more sensitive to distributional mismatches between the synthetically data and its pretraining data, or more training epochs may be needed for Recover-LoRA to be effective.

6.2 Parameter & Data Efficiency

Figure 2 shows the amount of training data used by each recovery method, for each model, to achieve the $AR\%$ in Table 2. Specifically, Recover-LoRA and LLM QAT* utilize 90k synthetic samples for AMD-OLMO-SFT, and 120k samples for all other models. In contrast, SFT LoRA utilizes a fixed, labeled dataset of 3M samples, previously selected by prior work to improve commonsense reasoning and multi-knowledge task performance (see Sec. 5).

Overall, we observe that Recover-LoRA achieves high $AR\%$ with less trainable parameters than LLM QAT* and less data than SFT-LoRA.

6.3 Synthetic Datasets in Recover-LoRA

The synthetic datasets utilized in Recover-LoRA enable a data-independent functional model degradation recovery method where good quality labeled data are not needed for training. Figure 3 shows that Recover-LoRA uses a minimum of 90k samples for positive $AR\%$ in three of the four models, with more data yielding higher $AR\%$. We hypothesize that applying Recover-LoRA to larger models will require more synthetic data. But more importantly, we show the flexibility of using synthetic data in Recover-LoRA, and that, depending on the application, such synthetic data can be readily generated and utilized.

6.4 Practical Development & Usage

While Recover-LoRA demonstrates strong accuracy recovery and efficiency across AMD-OLMO-SFT 1B, LLaMA3.2 1B and DeepSeek-R1-Distill-Qwen 1.5B, below we present several considerations for practical deployment. First, adapter placement can significantly impact recovery performance. As shown in Table 2, some models benefit from LoRA adapters on K and V projection layers, while other models benefit from adapters on all attention and MLP layers. Therefore, for practical deployment, a systematic search is necessary for identifying optimal adapter configurations per model. Additionally, the choice of model used for synthetic data generation influences recovery effectiveness. Specifically, Recover-LoRA works best when synthetic data is generated from a pretrained SLM that shares the same vocabulary and tokenizer as the degraded SLM. We provide these details in Appendix A. Also, a practical challenge posed with Recover-LoRA is diagnosing scenarios of limited recovery. Such limitations may stem from factors including suboptimal LoRA adapter configuration, insufficient synthetic data, or from architectural constraints of the degraded model itself. In the latter case, Recover-LoRA may be fundamentally limited in its ability to restore performance. Understanding these distinctions is crucial for maximizing Recover-LoRA’s effectiveness and ensuring its practical usability across models.

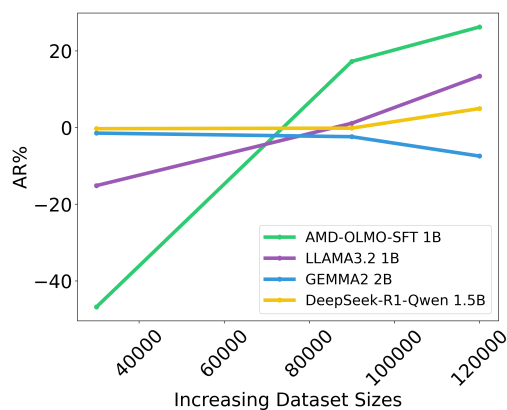


Figure 3: Progression of AR% with increasing dataset size, showing a minimum of 90k synthetic data samples are needed positive AR% in three models.

7 Conclusion

We introduce Recover-LoRA, a lightweight, dataset agnostic method to recover degraded model accuracy. Recover-LoRA leverages synthetic data to train LoRA adapters by using logit distillation to align a functionally degraded model with its pretrained SLM. Recover-LoRA does not require knowledge of the type of functional degradation. In this manner, Recover-LoRA provides a practical solution for recovering model degradation without requiring full model retraining or access to labeled data. Our results show the efficacy of Recover-LoRA in improving degraded model accuracies by 5-17%, while showcasing its parameter and data efficiency, highlighting its use case for real-world deployment.

Limitations Our results show Recover-LoRA to be effective for some MHA and GQA architectures. We also highlight that LoRA adapters are model-dependent in Recover-LoRA. Future work should investigate expanding the capabilities of Recover-LoRA to MQA (Shazeer, 2019), MLA (Liu et al., 2024) architectures; and how to automatically select the minimal set of LoRA adapters needed per model architectures using methods like Neural Architecture Search (NAS) (Ren et al., 2021). Additionally, future work should examine the applicability of Recover-LoRA on larger language models ranging between 7B-13B. Lastly, more experiments are needed to study the generalizability of Recover-LoRA in recovering accuracy from other sources of accuracy degradation such as quantization and pruning.

References

- Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.
- Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. 2021. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–32.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.
- Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, and 1 others. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.
- Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Purvish Jajal, Wenxin Jiang, Arav Tewari, Erik Kocinare, Joseph Woo, Anusha Sarraf, Yung-Hsiang Lu, George K Thiruvathukal, and James C Davis. 2023. Analysis of failures and risks in deep learning model converters: A case study in the onnx ecosystem. *arXiv preprint arXiv:2303.17708*.
- Wenjin Ke, Zhe Li, Dong Li, Lu Tian, and Emad Barsoum. 2025. DI-qat: Weight-decomposed low-rank quantization-aware training for large language models. *arXiv preprint arXiv:2504.09223*.
- Jiedong Lang, Zhehao Guo, and Shuyu Huang. 2024. A comprehensive study on quantization techniques for large language models. In *2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC)*, pages 224–231. IEEE.
- Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.
- Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, and 1 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.
- Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*.
- Nikolaos Loulidakis, Perry Gibson, José Cano, and Ajitha Rajan. 2023. Fix-con: Automatic fault localization and repair of deep learning model conversions between frameworks. *arXiv preprint arXiv:2312.15101*.
- Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D Lane, and Mengwei Xu. 2024. Small language models: Survey, measurements, and insights. *arXiv preprint arXiv:2409.15790*.
- Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. 2025. A survey on lora of large language models. *Frontiers of Computer Science*, 19(7):197605.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.
- Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. 2021. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 54(4):1–34.
- Bitu Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, and 1 others. 2023. Microscaling data formats for deep learning. *arXiv preprint arXiv:2310.10537*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need, 2019. *URL https://arxiv.org/abs*.
- Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and 1 others. Lora: Low-rank adaptation of large language models.
- Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Ameya Sunil Mahabaleshwarkar, Gerald Shen, Jiaqi Zeng, Zijia Chen, Yoshi Suhara, Shizhe Diao, and 1 others. 2024. Llm pruning and distillation in practice: The minitron approach. *arXiv preprint arXiv:2408.11796*.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Vithursan Thangarasa, Ganesh Venkatesh, Mike Lasby, Nish Sinnadurai, and Sean Lie. 2024. Self-data distillation for recovering quality in pruned large language models. *arXiv preprint arXiv:2410.09982*.
- Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. 2023. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*.
- Ofir Zafir, Ariel Larey, Guy Boudoukh, Haihao Shen, and Moshe Wasserblat. 2021. Prune once for all: Sparse pre-trained language models. *arXiv preprint arXiv:2111.05754*.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

A Synthetic Data Collection Hyperparameters

The hybrid sampling strategy in LLM QAT (Liu et al., 2023) generates the first few tokens greedily and the remaining tokens stochastically. In our usage of hybrid sampling, we set the number of greedily generated tokens to 5, and allow stochastic generation up to a max sequence length of 2048. The pretrained SLM is selected such that its vocabulary and tokenizer match with the degraded SLM to allow for meaningful training in Recover-LoRA. For example, our degraded Llama3.2 1B model utilized the pretrained Llama3.2 1B model for data generation. Our degraded Deepseek-R1-Distill-Qwen 1.5B model utilized the pretrained Llama3.2 1B model for data generation, since the Deepseek-R1-Distill-Qwen models utilize the LlamaFastTokenizer. Similarly, our degraded Gemma2 2B model utilized the pretrained Gemma2 2B model for data generation, and our degraded AMD-OLMO-SFT 1B model utilized the pretrained AMD-OLMO-SFT 1B model for data generation. Section 6 demonstrates the utility of hybrid sampling in recovering degraded model accuracy and details our ablation studies on the amount of synthetic data needed.

B Finetuning Hyperparameters

We performed a traditional hyperparameter sweep to select optimal hyperparameters for our Recover-LoRA method, traditional SFT LoRA baseline, as well as LLM QAT* baseline.

For Recover-LoRA and SFT LoRA we utilized a learning rate of $5e-4$, LoRA rank size of 64, LoRA alpha of 64, batch size of 1 with gradient accumulation of 32, a linear scheduler with 80 warm-up steps. Specifically for Recover-LoRA, we trained for 3 epochs, and for SFT LoRA we trained for 24k steps.

For LLM QAT* we utilized a learning rate of $2e-5$, a batch size of 1 with gradient accumulation of 32, a linear scheduler with 80 warmup steps and trained for 3 epochs.

C Evaluation Model Details

We evaluated Recover-LoRA on four different models, including Gemma2 2B (Team et al., 2024), Llama3.2 1B (Grattafiori et al., 2024), DeepSeek-R1-Distill-Qwen 1.5B (Guo et al., 2025) and AMD-

Olmo-SFT 1B³. These models were strategically chosen, given their different attention mechanisms: group-query attention (GQA) (Ainslie et al., 2023), and multi-head attention (MHA) (Chaudhari et al., 2021). In MHA, each head has its independent query, key and value projections, whereas in GQA, designated groups share the key and value projections. The AMD-OLMO-SFT 1B employs MHA, whereas Gemma2 2B model, Llama3.2 1B and DeepSeek-R1-Distill-Qwen 1.5B employ GQA.

³ <https://huggingface.co/amd/AMD-OLMo-1B-SFT>