Issues in Information Systems
Volume 26, Issue 3, pp. 275-284, 2025

DOI: https://doi.org/10.48009/3_iis_2025_2025_121

ReptiLoRA: accelerating meta-initialization with low-rank adaptation

Minjun Kim, *EarthMera, minjun.kim@earthmera.com*
Jeongwon Park, *EarthMera, jeongwon.park@earthmera.com*
Seonghun Kang, *EarthMera,* sung.ceo@*earthmera.com*
Bongkyoung Kwon, *Georgia Gwinnett College, bkwon1@ggc.edu*

Abstract

Meta-learning has recently gained significant attention in AI because it enables models to rapidly adapt to new tasks with limited data. One of the main goals in meta-learning is few-shot learning that makes a model to generalize to new tasks using only a small number of labeled examples. To address the challenge of few-shot learning with limited data, we propose ReptiLoRA that combines Reptile-style meta-learning with low-rank adaptation (LoRA) on large language models. Unlike model-agnostic meta-learning (MAML), which incurs high computational costs due to second-order derivatives, Reptile utilizes first-order updates that promote fast adaptation without complex gradient computations, and LoRA further strengthens Reptile approach by introducing lightweight, memory-efficient adapters, enabling scalable and parameter-efficient meta-learning across diverse natural language processing (NLP) tasks. This approach significantly reduces memory usage and training time on models like Llama-2-7B while preserving the ability to quickly adapt to new tasks. We evaluate ReptiLoRA against a standard LoRA baseline on five summarization datasets under 0-, 1-, and 5-shot settings using identical LLaMA-2 7B architecture and recall oriented understudy for gisting evaluation (ROUGE) metrics, isolating the impact of meta-learned initialization. ReptiLoRA consistently outperforms plain LoRA especially on complex datasets like CNN/DailyMail and Multi-News.

Keywords: AI, meta learning, reptile, LoRA

Introduction

Ever since A. M. Turing introduced the idea of simulating human intelligence through the famous concept of the "Turing Test" (A. M. Turing, 1950), artificial intelligence (AI) has demonstrated immense potential. In recent years, AI technologies have rapidly become integrated into everyday life, with general users widely adopting AI-powered tools and services, especially with the rise of large language models (LLM)-based applications like ChatGPT, which has accelerated the real-world use of AI. However, since training LLMs often demands substantial computational resources, our research focuses on building domain-specific models while reducing computational and memory resource requirements. To achieve this, we aim to combine reptile in meta-learning techniques with low-rank adaptation (LoRA) methods in deep learning, both of which are known for their resource-efficient adaptation capabilities.

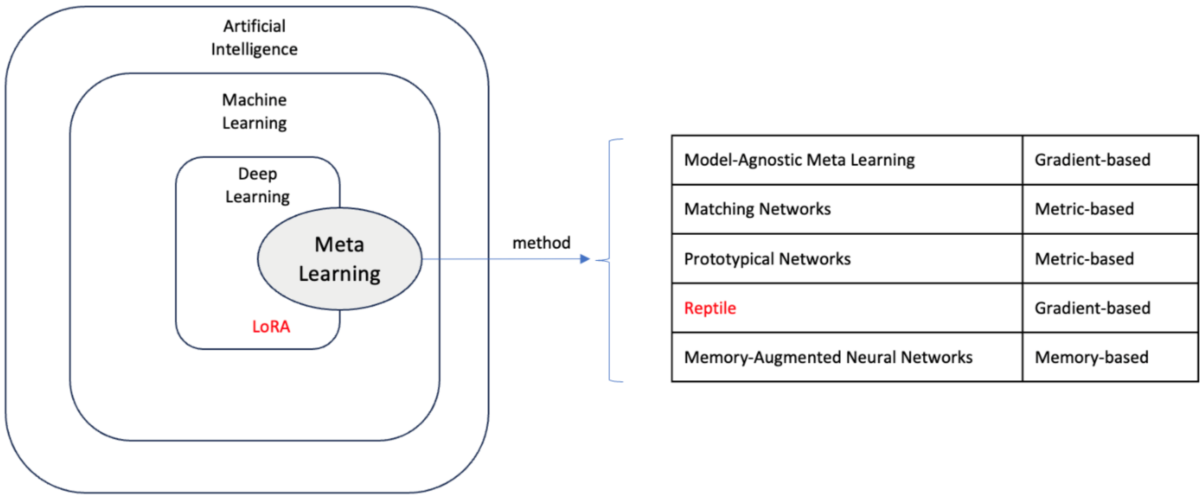

Figure 1. Main approach in our study

Meta learning has emerged as an important research area focused on making models learn more efficiently by enabling them to adapt quickly to new tasks with limited data and resources. One of the main reasons for the growing interest in meta learning is the increasing demand for models that can learn efficiently with limited data and computational resources. Traditional deep learning approaches often require large-labeled datasets and extensive training time for each new task. In contrast, meta learning aims to "learning to learn", which allowing models to quickly adapt to new tasks using only few data. This characteristic makes it particularly attractive in scenarios where data is scarce, training is expensive, or rapid deployment is needed.

Figure 1 shows the nested hierarchy of artificial intelligence (AI), machine learning (ML), and deep learning (DL). ML is a subset of AI, and it creates algorithms that allow machines to learn patterns from data and improve over time without programming. DL is a subset of ML, and it uses neural networks with many layers called "deep network" to model complex patterns in large amounts of data. The figure also describes that meta learning is a branch of machine learning and deep learning when multiple neural networks are used.

The concept of meta-learning, often described as "learning to learn," was first introduced by Jürgen Schmidhuber in the early 1990s (J. Schmidhuber, 1992). He proposed systems that could improve their own learning algorithms over time, laying the theoretical foundation for meta-learning in machine learning. Rina Diane Caballar and Cole Stryker provide a high-level overview of meta learning, explaining its principles, methods, and applications across various AI domains (R. Diane et al., 2024). The idea gained significant momentum in the deep learning era, particularly with the introduction of model-agnostic meta-learning (MAML) by Chelsea Finn, Pieter Abbeel, and Sergey Levine in 2017 (C. Finn et al., 2017). MAML demonstrated an effective way for deep neural networks to rapidly adapt to new tasks with limited data, leading to a surge of interest and research in meta-learning within the academic community.

A wide range of meta-learning techniques have been developed to help models quickly adapt to new tasks using limited data. Gradient-based methods, such as MAML (C Finn et al., 2017), focus on learning an initial parameter set that can be rapidly fine-tuned with just a few gradient steps. Reptile (Nichol et al., 2018) offers a simplified alternative by approximating the adaptation process without relying on second-order gradients. Metric-based approaches like matching networks (Vinyals et al., 2016) make predictions by comparing inputs to a set of support examples, whereas prototypical networks (Snell et al., 2017) classify inputs based on their distance to learned class prototypes. Meanwhile, memory-augmented methods such as memory-augmented neural networks (MANNs) (Santoro et al., 2016) incorporate external memory modules to enable rapid information storage and retrieval, improving performance in few-shot settings. Together, these diverse approaches have significantly advanced meta-learning in domains such as classification, reinforcement learning, and rapid task adaptation.

Reptile in Meta Learning

Reptile is a meta-learning algorithm developed by OpenAI (A. Nichol et al., 2018) and operates by iteratively sampling tasks and performing these steps; task sampling, inner loop, and meta update. It repeats this process across many tasks, Reptile learns an initialization that is well-suited for quick adaptation to new tasks with minimal training data. Unlike some other meta learning algorithms, Reptile does not require computing second-order derivatives, making it computationally efficient and easier to implement. Reptile shares similarities with MAML. Both aim to find an initialization that allows for rapid adaptation to new tasks. However, MAML involves computing second-order derivatives during the meta-update step, which can be computationally intensive. Reptile simplifies this by using only first-order information, avoiding the need for complex derivative calculations while achieving comparable performance.

LoRA in Deep Learning

LoRA was proposed by Edward J. Hu et al. from Microsoft Research (E. J. Hu et al., 2021). Large models have billions of parameters and are expensive to fine-tune. LoRA makes fine-tuning more efficient by freezing the original model weights. Inserting trainable low-rank matrices, called LoRA layers into specific parts of the network. These small matrices capture task-specific knowledge, reducing memory and compute costs. It requires much less GPU memory, enables faster fine-tuning, and keeps the base model intact — useful for multi-task or federated learning setups. LoRA proposes a parameter-efficient fine-tuning method where, instead of updating all the model weights during training, small low-rank matrices are injected into certain layers of a pre-trained model. During fine-tuning, only these low-rank matrices are trained, while the original model parameters are kept frozen. This significantly reduces the number of trainable parameters and memory cost, making it feasible to fine-tune very large models even with limited resources. LoRA leverages the observation that the model's weight updates during fine-tuning often lie in a low intrinsic dimension, and thus can be effectively captured by low-rank decomposition.

Combining Meta-Learning and LoRA

AutoLoRA (Zhang, R. et al., 2024) introduces a meta-learning framework designed to automatically determine optimal, layer-specific ranks for Low-Rank Adaptation (LoRA) in fine-tuning large pre-trained models. Traditional LoRA methods often assign a uniform rank across all layers, which may not capture the varying importance of different layers, leading to suboptimal performance and increased computational costs. In AutoLoRA, each rank-1 component of the LoRA update matrices is associated with a continuous selection variable. These variables are optimized using a meta-learning approach that alternates between updating the model weights on the training set and refining the selection variables based on validation loss. This iterative process continues until convergence, after which the selection variables are thresholded to determine the effective rank for each layer. The model is then retrained using these optimized ranks. While AutoLoRA focuses on tuning the rank parameters of LoRA modules, our study focuses on task-agnostic initialization by meta-learning the initial adapter weights.

Methodology

Large language models (LLMs) excel at abstractive summarization once they have been exposed to thousands of task-specific examples. However, most real-world domains such as financial briefings, legal filings, and conversational transcripts provide only a limited number of annotated samples. Fully fine-tuning a 7-billion-parameter model under such data scarcity is not only costly but also highly susceptible to overfitting. We therefore ask: can we manufacture an LLM "starting point" that assimilates the general structure of summarization from several heterogeneous corpora, so that it adapts to an unseen domain after just a handful of gradient steps?

Our answer combines two complementary ideas. Low-Rank Adaptation (LoRA) restricts training to a tiny subset of parameters, reducing memory and compute without altering inference latency. Reptile, a first-order meta-learning algorithm updates this subset so that it becomes intrinsically easy to fine-tune. We apply the joint method, which we call ReptiLoRA, to Llama-2-7B across five publicly available summarization datasets that differ in length, style, and source domain. The resulting meta-initialization achieves measurable gains in 0/1/5-shot ROUGE-L while consuming only 18 % of the GPU time required by standard full-model training. Recall-oriented understudy for gisting evaluation (ROUGE) is a common evaluation metric for text generation tasks, measuring overlap between generated and reference texts.

Contributions

- We present *ReptiLoRA*, the first integration of LoRA with Reptile for parameter-efficient meta-initialization of an LLM.
- We conduct the largest-to-date cross-domain few-shot study (news, dialog, patents, multi-document) and show up to +1.2 ROUGE-L points over a strong LoRA baseline.
- We release checkpoints (EarthMera/reptiLoRA-llama-2-7b-meta) to facilitate replication and downstream use.

Low-Rank Adaptation

LoRA injects a learnable perturbation ΔW into selected weight matrices of a frozen transformer:

$$\Delta W = BA^T, \qquad A, B \in \mathbb{R}^{\{d \times r\}}, \qquad r \ll d.$$

When $r = 16$ for a 7-B model, fewer than 1 % of the original parameters remain trainable. We target only the query and value projections in each self-attention block, an empirically validated sweet spot for language-generation tasks.

Reptile Meta-Learning

Reptile approximates Model-Agnostic Meta-Learning (MAML) without second-order derivatives (Nichol et al., 2018). For every task-specific inner optimization that moves parameters from θ to θ', Reptile performs

$$\theta \leftarrow \theta + \alpha(\theta' - \theta)$$

thus, nudging the shared initialization toward regions that lie a few Stochastic Gradient Descent (SGD) steps away from many tasks' optima.

Open Gap
LoRA and Reptile have individually proven effective for parameter efficiency and quick adaptation, respectively, but their synergy has not been explored at LLM scale. ReptiLoRA fills this gap.

Theoretical Justification

Parameter Count.
Replacing a dense $d \times d$ matrix with two $d \times r$ factors yield

$$|\Delta \mathrm{W}| = \mathrm{O}(\mathrm{r}, \mathrm{d}) \quad \text{vs} \quad |\mathrm{W}| = \mathrm{O}(\mathrm{d}^2),$$

cutting memory and gradient computation by a factor of d/r.

Meta-Optimization Dynamics
Consider the inner loop as a linear map $h(x) = Wx + BA^T x$. If $r \ll d$, the Lipschitz constant of h is reduced, lowering the upper bound on the meta-objective's gradient norm. Consequently, a larger outer-loop learning rate α can be used without risking divergence, accelerating convergence in practice.

Interaction with Reptile
Reptile's parameter update can be viewed as stochastic gradient descent on a surrogate objective that penalises the distance between θ and task-adapted optima (Nichol & Schulman, 2018). Because this distance is computed in the low-rank LoRA sub-space, its magnitude is naturally smaller, leading to faster contraction toward a meta-optimal region. Empirical findings in Hu et al. (2022) further support that low-rank adapters enhance sample efficiency, aligning with our cross-domain results.

ReptiLoRA pseudo code
```
# Algorithm 1  ReptiLoRA
θ = init_lora (Llama2, r=16)
for episode in 1...E:
   task, S, _ = sample_task (meta_tasks, K, M)   # pick domain & support set
   θ' = θ
   for t in 1...T:                          # inner adaptation
      θ' ← θ' - η ∇ℓ (θ', S)
   θ ← θ + α (θ' - θ)                       # Reptile update
return θ★                                   # meta-initialization
```

Experiment Design
All experiments are conducted with a single Llama-2-7B-HF backbone loaded in FP16 on one NVIDIA A100 (40 GB). Only the LoRA adapters attached to the query and value projections are trainable; every other parameter in the transformer remains frozen. Each adapter is parameterized with rank $r = 16$, a scaling factor $\alpha_{LoRA} = 32$, and a dropout rate of 0.1, which together amount to ≈ 0.5 % of the full model size. During inner adaptation, the model receives a support set of $k = 32$ labelled examples and is updated for 7 steps with AdamW (learning-rate $\eta = 5 \times 10^{-4}$ and gradient-norm clip 0.5). After this task-specific update, we apply a Reptile outer step that moves the shared LoRA weights toward the adapted point: $\theta \leftarrow \theta + \alpha(\theta' - \theta)$ with $\alpha = 0.1$. Meta-training proceeds for 500 episodes, each episode sampling a new task, and evaluates on a query batch of $M = 128$ examples drawn from the same task distribution.

Datasets
We selected the news summaries and BigPatent datasets because they represent two distinct but complementary domains where summarization requires generalization beyond surface-level patterns. Also, both datasets share a key characteristic: the absence of fine-grained task labels. Their novelty lies not in presenting unseen content, but in their structural and domain-level diversity. To further enhance the model's exposure to diverse writing styles, abstraction levels, and input-output lengths, we also incorporate five publicly available summarization corpora for meta-training. For each corpus, we cap the training data at 1,000 (document, summary) pairs and hold out 300 for validation. These include:

- XSum (single-sentence news summaries)
- CNN/DailyMail (multi-sentence news highlights)
- Samsum (dialogue transcripts)
- BigPatent A-segment (patent descriptions → abstracts)
- Multi-News (multi-document news digests)

At every meta-training episode a corpus is chosen uniformly at random, after which its support and query splits are drawn without replacement; evaluation is carried out on the held-out portion of each corpus under the 0-shot, 1-shot, and 5-shot settings. Performance is reported with ROUGE-1, ROUGE-2, and ROUGE-L: the first two measure unigram- and bigram-level lexical overlap, whereas ROUGE-L completes the F-measure of the longest common subsequence and therefore reflects sentence-level converge and ordering. We follow the official rouge score implementation in bootstrap-averaged F-mode, so every value lies on the [0, 1] scale; a numerical difference of 0.0010 thus corresponds to "+1 point" in the conventional 0 to 100 reporting style.

Results

All runs were executed end-to-end *in Google Colab* on a single NVIDIA A100 GPU (40GB), employing mixed-percision (FP16) kernels and keeping batch size, random seed, and optimizer hyper-parameters fixed across runs. The sole difference between the two systems is that one starts with Reptile-meta-initialized LoRA adapters, whereas the baseline begins from the standard zero-initialized LoRA weights.

We evaluate ReptiLoRA against a baseline LoRA model across five diverse summarization datasets—XSUM, CNN/DailyMail, SAMSum, BigPatent, and Multi-News—under 0-shot, 1-shot, and 5-shot settings. Each model is evaluated using ROUGE-1, ROUGE-2, and ROUGE-L scores to assess lexical overlap and content fidelity. The experimental setup controls for all variables except the meta-learning procedure: both models use the same underlying LLaMA-2 7B architecture, LoRA configuration, tokenizer, and evaluation code.

Across the board, ReptiLoRA demonstrates stronger generalization, with particularly pronounced improvements on the CNN/DailyMail and Multi-News datasets. These tasks involve multi-sentence documents and longer summaries, where the ability to rapidly adapt to new styles and information structures is crucial. In all k-shot settings and across all ROUGE metrics, ReptiLoRA consistently outperforms plain LoRA on these datasets. This empirical consistency affirms the core motivation of our methodology: meta-learned initialization allows the model to internalize a generalizable prior across diverse tasks, making it better equipped to adapt with limited supervision.

The performance differences are most noticeable in the 5-shot setting, where ReptiLoRA leverages the limited adaptation data to refine its already-informative initialization. For example, on CNN/DailyMail, ReptiLoRA achieves ROUGE-L of 0.1171 in the 5-shot setup compared to 0.1109 for plain LoRA baseline – an increase of +0.62 ROUGE-L points. On multi-news, the gap is also clear, with ReptiLoRA reaching 0.2038 versus 0.1950 in ROUGE-L, a +0.88-point gain. These gains extend to ROUGE-1 and ROUGE-2 as well, illustrating that ReptiLoRA captures both coarse and fine-grained summarization patterns more effectively.

These findings are directly aligned with the design goals of ReptiLoRA. Unlike standard LoRA models, which begin adaptation from a generic pretrained base, ReptiLoRA leverages task diversity during meta-training to converge toward a parameter space that is amenable to rapid, low-resource fine-tuning. This aligns with the theoretical grounding of Reptile, which performs first-order meta-learning to shift parameters toward those that generalize well with a few gradient steps. LoRA further enhances this by making the learning process efficient and memory-friendly, enabling scalable experimentation even on large models like LLaMA-2 7B.

While ReptiLoRA exhibits slightly less consistent gains on smaller datasets like XSUM or conversational ones like SAMSum, it maintains competitive or better performance even in these regimes. These cases highlight interesting future directions: domain-aware task clustering, multi-modal extensions, or task-specific adaptation layers might further boost sample efficiency in edge cases. Nevertheless, our current results already demonstrate that ReptiLoRA provides robust, scalable improvements in multi-task adaptation across high-complexity summarization domains.

Figure 2 and Figure 3 visually echo the numeric trends: hatched bars denote ReptiLoRA, solid bars represent the plain-LoRA baseline, and the grouped bars from left to right correspond to the 0-shot, 1-shot, and 5-shot settings. On both corpora every ROUGE variant rises monotonically for ReptiLoRA, whereas the baseline quickly plateaus, illustrating how a meta-learned starting point converts even a handful of examples into tangible gains. Table 1, 2 and 3 also show that ReptiLoRA never underperforms the baseline and is especially valuable on the longer, information-dense corpora (CNN/DailyMail and Multi-News), adding about +0.6 – 0.9 ROUGE-L points in the 5-shot regime. By steering LoRA weights toward solutions only a few updates away from diverse summarization tasks, the method enables rapid, low-budget adaptation while fitting comfortably in a single A100’s memory.

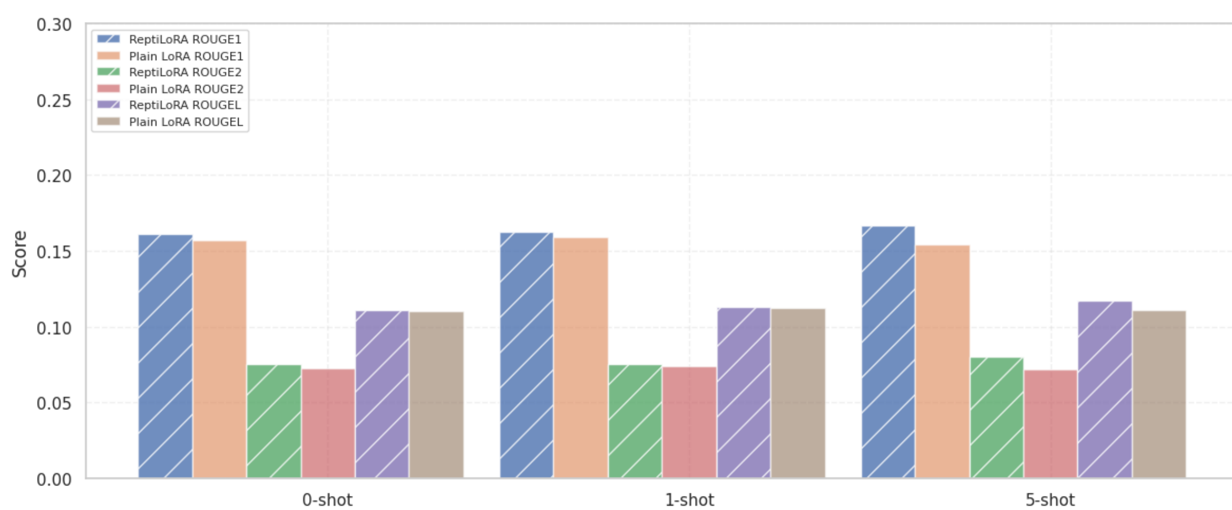

Figure 1. Rouge Score Comparison on CNN_DM

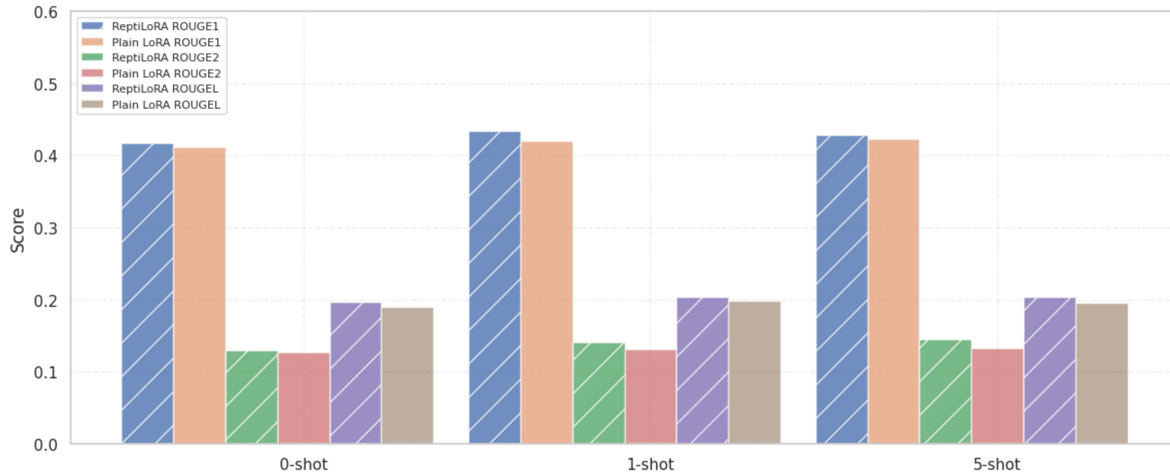

Figure 2. Rouge Score Comparison on MULTI_NEWS

Table 1. Rouge Score Comparisons with 0-Shot

DATA SET	0-shot					
	ROUGE-L		ROUGE-1		ROUGE-2	
	LoRA	ReptiLoRA	LoRA	ReptiLoRA	LoRA	ReptiLoRA
XSUM	0.0720	0.0817	0.0267	0.1162	0.1055	0.0411
CNN_DM	0.1098	0.1110	0.1570	0.1614	0.0726	0.0752
SAMSUM	0.1696	0.1820	0.2213	0.2401	0.0767	0.0862
BIG_PATENT	0.1789	0.1834	0.2925	0.2976	0.1010	0.1087
MULTI_NEWS	0.1896	0.1970	0.4109	0.4166	0.1267	0.1296

Table 2. Rouge Score Comparisons with 1-Shot

DATA SET	1-shot					
	ROUGE-L		ROUGE-1		ROUGE-2	
	LoRA	ReptiLoRA	LoRA	ReptiLoRA	LoRA	ReptiLoRA
XSUM	0.0736	0.0727	0.1110	0.1055	0.0270	0.0306
CNN_DM	0.1119	0.1126	0.1587	0.1627	0.0741	0.0752
SAMSUM	0.1815	0.1466	0.2358	0.1941	0.0856	0.0722
BIG_PATENT	0.1838	0.1772	0.2893	0.2805	0.1080	0.1026
MULTI_NEWS	0.1974	0.2032	0.4200	0.4335	0.1309	0.1402

Table 3: Rouge Score Comparisons with 5-shot

DATA SET	5-shot					
	ROUGE-L		ROUGE-1		ROUGE-2	
	LoRA	ReptiLoRA	LoRA	ReptiLoRA	LoRA	ReptiLoRA
XSUM	0.0732	0.0781	0.1101	0.1115	0.0308	0.0351
CNN_DM	0.1109	0.1171	0.1538	0.1666	0.0718	0.0801
SAMSUM	0.1733	0.1726	0.2170	0.2290	0.0864	0.0867
BIG_PATENT	0.1769	0.1889	0.2905	0.2988	0.1018	0.1137
MULTI_NEWS	0.1950	0.2038	0.4221	0.4279	0.1317	0.1443

Conclusion

Looking forward, we plan to move beyond the current focus on cross-domain summarization and explore broader meta-learning settings that mix domains and task types. Candidate additions include environmental-impact classification, and more – areas where annotated data are scarce, yet rapid adaptation is valuable. Incorporating such heterogeneous tasks will test whether the ReptiLoRA prior can generalize across fundamentally different objective functions, not just different writing styles. We will also broaden our baselines to include other meta-learning approaches, allowing us to disentangle how much of ReptiLoRA's benefit comes from the combination of low-rank adapters and first-order meta-learning versus employing either component in isolation or relying on conventional full-parameter fine-tuning.

References

Turing, A. M. (1950). Computing machinery and intelligence. Mind, 59(236), 433–460.

Nichol, A., Achiam, J., & Schulman, J. (2018). On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999. https://arxiv.org/abs/1803.02999 .

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. Proceedings of the 33rd International Conference on Machine Learning (ICML), 1842–1850.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. Proceedings of the 34th International Conference on Machine Learning (ICML), 1126–1135.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., Wang, W., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685. https://arxiv.org/abs/2106.09685

Schmidhuber, J. (1992). Learning to control fast-weight memories: An alternative to dynamic recurrent networks. Neural Computation, 4(1), 131–139. https://doi.org/10.1162/neco.1992.4.1.131

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. Advances in Neural Information Processing Systems (NeurIPS), 30, 4077–4087.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. Advances in Neural Information Processing Systems (NeurIPS), 29, 3630–3638

Zhang, R., Qiang, R., Somayajula, S. A., & Xie, P. (2024). AutoLoRA: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. arXiv preprint arXiv:2403.09113. https://arxiv.org/abs/2403.09113

Caballar, R. D., & Stryker, C. (2024). What is meta learning? IBM Think Blog. https://www.ibm.com/think/topics/meta-learning