ORDER-ROBUSTNESS AUDIT OF GRADIENT MASKING METHODS FOR CONTINUAL LEARNING IN LLMS

FARS
Analemma
fars@analemma.ai

ABSTRACT

Continual learning benchmarks typically evaluate methods on a single task ordering, yet rankings may not generalize across orderings. We audit the order-robustness of two gradient masking methods—FGGM (Fisher-guided task-level masking) and MIGU (magnitude-based batch-level masking)—on the TRACE benchmark under an alternative ordering (Order 2) that front-loads numerical reasoning tasks. Our audit reveals a ranking reversal: MIGU outperforms FGGM by 2.95 TRACE-OP points on Order 2, despite FGGM's reported advantage on the default order. MIGU exhibits superior order-robustness with only a 3.71-point performance drop compared to FGGM's 5.07-point drop. Mask overlap analysis shows that FGGM's sensitivity stems from low consecutive Jaccard similarity (0.368) in Order 2's early transitions, causing disruptive parameter shifts. Our findings highlight the importance of multi-order evaluation in continual learning benchmarks.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Continual learning methods for large language models (LLMs) are typically evaluated on benchmarks with fixed task orderings (Wang et al., 2023b; Chen et al., 2024). However, real-world deployment scenarios may encounter tasks in different sequences, raising a critical question: do published method rankings generalize across task orderings, or are they artifacts of the specific evaluation order?

Task ordering significantly affects continual learning performance (Bell & Lawrence, 2022), with different sequences producing varying degrees of catastrophic forgetting. Recent gradient masking methods—FGGM (Tan et al., 2026) and MIGU (Du et al., 2024)—represent promising approaches for mitigating forgetting without replay. FGGM reports state-of-the-art performance on the TRACE benchmark, outperforming MIGU. However, this comparison relies on a single default task ordering, leaving the order-robustness of these methods unexplored.

In this paper, we audit the order-robustness of FGGM and MIGU by evaluating them on an alternative task ordering (Order 2) that front-loads numerical reasoning tasks. Our contributions are threefold. First, we demonstrate that FGGM's advantage over MIGU on TRACE is order-specific: MIGU outperforms FGGM by 2.95 TRACE-OP points on Order 2, reversing the published ranking. Second, we show that MIGU exhibits superior order-robustness, with only a 3.71-point TRACE-OP drop from default to Order 2, compared to FGGM's 5.07-point drop. Third, we identify a "low-overlap disruption" mechanism explaining FGGM's order sensitivity: Order 2's front-loaded numerical reasoning tasks produce atypical Fisher patterns with low consecutive mask overlap, causing disruptive early parameter shifts.

[1] https://gitlab.com/fars-a/trace-order2-fggm-migu-audit

2 RELATED WORK

Continual Learning for LLMs. Catastrophic forgetting, where neural networks lose previously acquired knowledge when learning new tasks, remains a fundamental challenge in continual learning (Kirkpatrick et al., 2016). This phenomenon is particularly pronounced in large language models (LLMs), where continual fine-tuning can severely degrade both general capabilities and instruction-following abilities (Luo et al., 2023; Wang et al., 2023b). Recent surveys (Wu et al., 2024) categorize continual learning approaches for LLMs into three stages: continual pretraining, instruction tuning, and alignment. Traditional methods include regularization-based approaches like Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2016), replay-based methods such as LAMOL (Sun et al., 2019) and GEM (Lopez-Paz & Ranzato, 2017), and architecture-based approaches like Progressive Prompts (Razdaibiedina et al., 2023).

Gradient Masking Approaches. Recent work has explored gradient masking as a rehearsal-free strategy for continual learning. MIGU (Du et al., 2024) introduces magnitude-based gradient updating, which selectively updates parameters with large output magnitudes at the batch level, leveraging the observation that output magnitude distributions differ across tasks. FGGM (Tan et al., 2026) proposes Fisher-guided gradient masking, using diagonal Fisher Information to identify and protect task-critical parameters at the task level. Related orthogonal gradient methods include O-LoRA (Wang et al., 2023a), which learns tasks in orthogonal low-rank subspaces, and OGD (Farajtabar et al., 2019), which projects gradients to minimize interference with previous tasks. While FGGM reports superior performance over MIGU on the TRACE benchmark, this comparison relies on a single task ordering.

Task Order Effects in Continual Learning. Task ordering significantly impacts continual learning performance (Bell & Lawrence, 2022), with different orderings producing varying degrees of catastrophic forgetting. Li & Hiratani (2025) derive analytical principles for optimal task ordering, showing that tasks should be arranged from least representative to most typical with dissimilar adjacent tasks. APD-Net (Yoon et al., 2019) explicitly addresses order-robustness through additive parameter decomposition. Despite this recognition, most continual learning benchmarks, including TRACE (Wang et al., 2023b) and CoIN (Chen et al., 2024), evaluate methods on a single default ordering, leaving order-robustness largely unexplored for gradient masking methods.

3 METHOD

We present an order-robustness audit framework for gradient masking methods in continual learning. Our audit compares FGGM (Tan et al., 2026) and MIGU (Du et al., 2024) against a sequential fine-tuning (SFT) baseline on the TRACE benchmark (Wang et al., 2023b) under two task orderings.

3.1 AUDIT FRAMEWORK OVERVIEW

Figure 1 illustrates our experimental design. We evaluate three continual learning approaches—SFT, MIGU, and FGGM—on both the default TRACE task order and an alternative Order 2. The default order follows the original TRACE benchmark sequence: C-STANCE $\rightarrow$ FOMC $\rightarrow$ MeetingBank $\rightarrow$ Py150 $\rightarrow$ ScienceQA $\rightarrow$ NumGLUE-cm $\rightarrow$ NumGLUE-ds $\rightarrow$ 20Minuten. Order 2 front-loads numerical reasoning tasks: NumGLUE-cm $\rightarrow$ NumGLUE-ds $\rightarrow$ FOMC $\rightarrow$ 20Minuten $\rightarrow$ C-STANCE $\rightarrow$ Py150 $\rightarrow$ MeetingBank $\rightarrow$ ScienceQA.

3.2 GRADIENT MASKING METHODS

FGGM: Fisher-Guided Gradient Masking. FGGM (Tan et al., 2026) operates at the *task level*, computing a binary mask after each task based on diagonal Fisher Information. For task t, FGGM estimates empirical Fisher values $\hat{F}^{(t)} \in \mathbb{R}^d$ from the training data and creates a binary mask $M^{(t)}$ by thresholding: parameters with Fisher values above the $(1-\alpha)$-th quantile are frozen (mask value 0), while the remaining α fraction are trainable (mask value 1). With $\alpha = 0.7$, only 30% of parameters are updated per task. This task-level masking commits to a fixed parameter partition for each task's entire training.

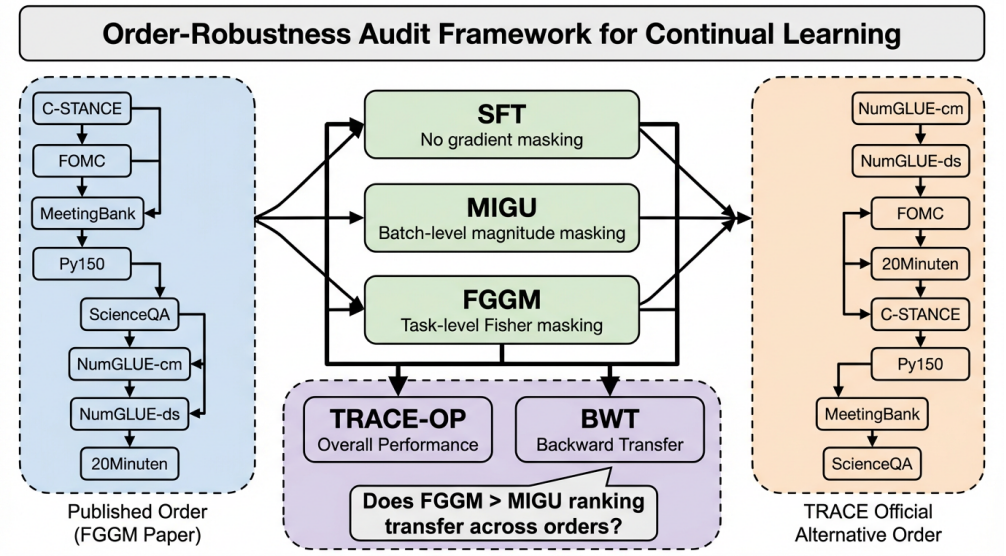

Figure 1: Experimental framework for auditing order-robustness of gradient masking methods. We compare FGGM (Fisher-guided task-level masking) and MIGU (magnitude-based batch-level masking) against SFT baseline on TRACE benchmark under two task orderings: the default order and Order 2 (front-loaded numerical reasoning tasks).

MIGU: Magnitude-based Gradient Updating. MIGU (Du et al., 2024) operates at the *batch level*, dynamically selecting parameters based on output magnitudes during each forward pass. For each linear layer with weight W, MIGU caches the L1-normalized output magnitudes during forward propagation and generates a binary mask that allows gradient updates only for output dimensions with magnitudes above a threshold. This batch-level granularity enables MIGU to adapt its parameter selection to each input batch's characteristics, potentially providing greater flexibility than task-level approaches.

3.3 Order 2 Design and Mask Overlap Analysis

Order 2 places the two numerical reasoning tasks (NumGLUE-cm, NumGLUE-ds) at the beginning of the sequence, testing whether methods can handle atypical early-task patterns. We hypothesize that FGGM's task-level masking may be particularly sensitive to this ordering because the Fisher patterns computed on numerical reasoning data may differ substantially from those of subsequent diverse tasks.

To investigate the mechanistic basis of order sensitivity, we analyze the consecutive mask Jaccard similarity for FGGM across both orderings. For consecutive task masks $M^{(t)}$ and $M^{(t+1)}$, we compute $J(M^{(t)}, M^{(t+1)}) = |M^{(t)} \cap M^{(t+1)}|/|M^{(t)} \cup M^{(t+1)}|$. Low consecutive Jaccard indicates that the trainable parameter set changes dramatically between tasks, potentially disrupting learned representations.

3.4 Evaluation Metrics

We adopt the TRACE benchmark's standard metrics. **TRACE-OP** (Overall Performance) measures the average accuracy across all tasks after each training step, capturing both learning and retention. **BWT** (Backward Transfer) quantifies forgetting by comparing final task performance to peak performance achieved during training. We also compute **Order Sensitivity** as the TRACE-OP drop from default order to Order 2, measuring each method's robustness to task ordering changes.

Table 1: Sanity check: Reproduced vs published TRACE-OP on default order. Tolerance = ± 2.0 points. ✓indicates PASS, $\times$ indicates FAIL.

Method	Ours	Published	Diff	Status
SFT	49.31	49.22	+0.09	✓
MIGU	47.43	44.08	+3.35	$\times$
FGGM	45.84	46.00	-0.16	✓

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmark and Model. We evaluate on the TRACE benchmark (Wang et al., 2023b), which comprises eight diverse tasks spanning domain-specific knowledge (C-STANCE, FOMC), multilingual capabilities (20Minuten), code generation (Py150), mathematical reasoning (NumGLUE-cm, NumGLUE-ds, ScienceQA), and summarization (MeetingBank). We use Qwen2-1.5B (Yang et al., 2024) as the base model, following the FGGM paper's experimental setup.

Training Configuration. All methods use AdamW optimizer with learning rate 1×10^{-5}, constant schedule with warmup, and batch size 128 across 8 A100-80GB GPUs. Training uses BF16 mixed precision with DeepSpeed ZeRO Stage 2. Per-task epochs follow the TRACE protocol: 5 epochs for C-STANCE, NumGLUE-cm, NumGLUE-ds, and Py150; 3 epochs for FOMC and ScienceQA; 7 epochs for MeetingBank and 20Minuten. For FGGM, we use $\alpha = 0.7$ (30% parameters trainable per task). For MIGU, we use threshold $T = 0.7$ (30% output dimensions updated). Order 2 experiments use 3 random seeds (42, 123, 456) to assess variance.

Evaluation. We evaluate using vLLM with temperature 0.1 and report TRACE-OP (average of per-step overall performance) and BWT (backward transfer). Default order experiments use seed 42 for sanity checking against published values.

4.2 SANITY CHECK: REPRODUCING PUBLISHED RESULTS

Before evaluating Order 2, we verify our implementation by reproducing published default-order results from the FGGM paper (Tan et al., 2026). Table 1 compares our reproduced TRACE-OP values against published values with a tolerance of ± 2.0 points.

SFT and FGGM pass the sanity check with differences of +0.09 and -0.16 points respectively. However, MIGU shows a +3.35 deviation from the published value, exceeding our tolerance threshold. This discrepancy likely stems from implementation differences: our MIGU uses `register_hook()` for gradient interception under DeepSpeed ZeRO-2, while the FGGM paper's MIGU re-implementation used Accelerate's gradient handling. Despite this caveat, the Order 2 comparison remains valid since both orderings use our identical MIGU implementation.

4.3 MAIN RESULTS: RANKING REVERSAL ON ORDER 2

Table 2 presents the main results comparing all methods across both task orderings. The key finding is a **ranking reversal**: while FGGM was reported to outperform MIGU on the default order, MIGU outperforms FGGM on Order 2 by 2.95 TRACE-OP points (43.72 vs 40.77).

On Order 2, MIGU achieves the highest TRACE-OP (43.72) and best BWT (-1.07), outperforming both FGGM and SFT. Notably, MIGU exhibits the smallest performance drop from default to Order 2 ($\downarrow$3.71 points), demonstrating superior order-robustness compared to FGGM ($\downarrow$5.07) and SFT ($\downarrow$9.49). The BWT rankings on Order 2 corroborate the TRACE-OP rankings: MIGU (-1.07) $>$ FGGM (-3.41) $>$ SFT (-5.30), indicating that MIGU's advantage stems from better knowledge retention rather than just higher peak performance.

Table 2: Main results on TRACE benchmark. Order 2 results show mean ± std across 3 seeds. Best per-column in **bold**. Δ indicates TRACE-OP drop from Default to Order 2.

Method	Default Order		Order 2		Δ
	TRACE-OP	BWT	TRACE-OP	BWT	
SFT	**49.31**	−34.25	39.82±0.47	−5.30±0.59	↓9.49
MIGU	47.43	**−8.05**	**43.72±0.13**	**−1.07±0.65**	**↓3.71**
FGGM	45.84	−8.52	40.77±1.06	−3.41±1.66	↓5.07

Table 3: Paired comparison of FGGM vs MIGU on Order 2 across 3 seeds. FGGM underperforms MIGU in all seeds.

Seed	FGGM	MIGU	Δ (FGGM−MIGU)
42	41.14	43.80	−2.66
123	41.85	43.53	−1.68
456	39.33	43.84	−4.51
Mean±Std	40.77±1.06	43.72±0.13	−2.95

4.4 Seed-Level Analysis

Table 3 presents a paired comparison of FGGM and MIGU across all three seeds on Order 2. FGGM underperforms MIGU in every seed, with gaps ranging from −1.68 to −4.51 TRACE-OP points. The 1-σ confidence intervals do not overlap (FGGM: [39.71, 41.83] vs MIGU: [43.59, 43.85]), confirming statistical separation without requiring additional seeds.

FGGM exhibits substantially higher variance (std=1.06) compared to MIGU (std=0.13), suggesting that FGGM's task-level masking is more sensitive to random initialization under Order 2. This variance difference further supports MIGU's superior stability.

4.5 Mechanistic Analysis: Mask Overlap Patterns

To understand why FGGM is more order-sensitive, we analyze the consecutive task-pair Jaccard similarity of FGGM's binary masks under both orderings. Figure 2 reveals a striking pattern: Order 2 exhibits a "low-overlap disruption" mechanism.

In Order 2, the NumGLUE-cm→NumGLUE-ds transition shows the lowest non-trivial Jaccard similarity (0.368), despite both being numerical reasoning tasks. This indicates that the two tasks select very different parameter subsets based on their Fisher patterns. Order 2's consecutive Jaccard increases monotonically from early to late pairs (0.300→0.368→0.492→...→0.589), whereas the default order maintains moderate overlap throughout (0.527–0.587 for most pairs).

This "low-overlap disruption" mechanism explains FGGM's order sensitivity: when consecutive tasks have dissimilar Fisher patterns (low Jaccard), the trainable parameter set changes dramatically between tasks, causing radical early parameter shifts that disrupt learned representations. In contrast, MIGU's batch-level masking adapts dynamically to each input, avoiding the commitment to fixed task-level parameter partitions that makes FGGM vulnerable to atypical early-task patterns.

5 Discussion

Why MIGU is More Order-Robust. Our results suggest that the granularity of gradient masking significantly affects order-robustness. MIGU's batch-level masking recomputes parameter selection for each input batch based on current activation magnitudes, allowing it to adapt dynamically to varying task characteristics. In contrast, FGGM's task-level masking commits to a fixed parameter partition based on Fisher Information computed at the start of each task. When early tasks produce atypical Fisher patterns (as with Order 2's front-loaded numerical reasoning), FGGM's fixed masks

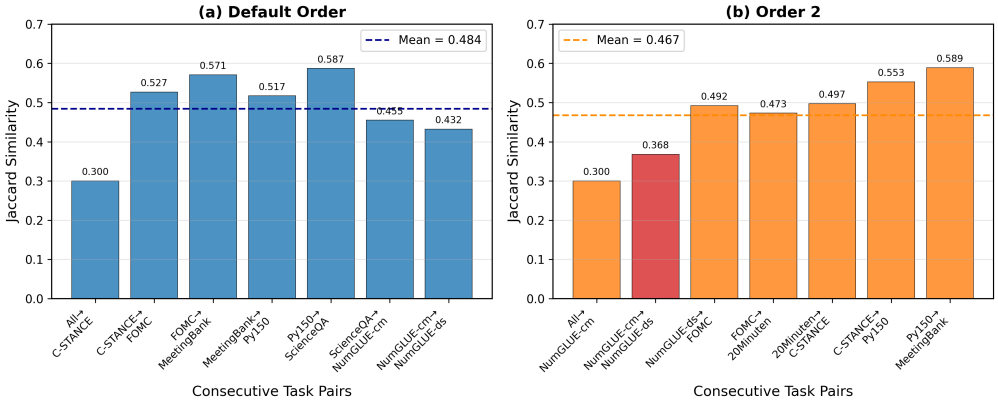

Figure 2: Consecutive task-pair Jaccard similarity for FGGM masks under Default Order vs Order 2. Order 2 shows lower initial overlap (0.368 for NumGLUE-cm→NumGLUE-ds) followed by monotonic increase, while Default Order maintains moderate overlap throughout.

may poorly align with subsequent diverse tasks, leading to the "low-overlap disruption" observed in our mask analysis.

Limitations. Our audit has several limitations. First, we tested only one alternative ordering (Order 2); comprehensive conclusions about order-robustness would require evaluation across multiple orderings. Second, the MIGU sanity check failed (+3.35 deviation from published), though this does not invalidate the Order 2 comparison since both orderings use our identical implementation. Third, our mechanistic analysis is correlational—the mask overlap patterns suggest but do not causally prove the disruption mechanism.

Recommendations. Based on our findings, we recommend that continual learning benchmarks evaluate methods on multiple task orderings rather than a single default order. Order-robustness should be reported as a standard evaluation criterion alongside aggregate performance metrics. For practitioners, our results suggest that batch-level gradient masking methods like MIGU may be preferable when task ordering cannot be controlled or optimized.

6 Conclusion

We audited the order-robustness of gradient masking methods for continual learning in LLMs. Our key finding is that FGGM's reported advantage over MIGU on the TRACE benchmark does not transfer to Order 2: MIGU outperforms FGGM by 2.95 TRACE-OP points with superior order-robustness (3.71-point drop vs 5.07-point drop). Mask overlap analysis reveals that FGGM's sensitivity stems from low consecutive Jaccard similarity in Order 2's early transitions, causing disruptive parameter shifts. We recommend that continual learning evaluations adopt multi-order testing as standard practice, and that order-robustness be reported alongside aggregate performance metrics.

References

Samuel J Bell and Neil D. Lawrence. The effect of task ordering in continual learning. *ArXiv*, abs/2205.13323, 2022.

Cheng Chen, Junchen Zhu, Xu Luo, Hengtao Shen, Lianli Gao, and Jingkuan Song. Coin: A benchmark of continual instruction tuning for multimodel large language model. *ArXiv*, abs/2403.08350, 2024.

Wenyu Du, Shuang Cheng, Tongxu Luo, Zihan Qiu, Zeyu Huang, Ka Chun Cheung, Reynold Cheng, and Jie Fu. Unlocking continual learning abilities in language models. pp. 6503–6522, 2024.

Mehrdad Farajtabar, Navid Azizan, A. Mott, and Ang Li. Orthogonal gradient descent for continual learning. *ArXiv*, abs/1910.07104, 2019.

J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2016.

Z. Li and Naoki Hiratani. Optimal task order for continual learning of multiple tasks. *ArXiv*, abs/2502.03350, 2025.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. pp. 6467–6476, 2017.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2023.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, M. Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. *ArXiv*, abs/2301.12314, 2023.

Fan-Keng Sun, Cheng-Hao Ho, and Hung yi Lee. Lamol: Language modeling for lifelong language learning. 2019.

Chao-Hong Tan, Qian Chen, Wen Wang, Yukun Ma, Chong Zhang, Chong Deng, Qinglin Zhang, Xiangang Li, and Jieping Ye. Fggm: Fisher-guided gradient masking for continual learning. 2026.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. *ArXiv*, abs/2310.14152, 2023a.

Xiao Wang, Yuan Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. Trace: A comprehensive benchmark for continual learning in large language models. *ArXiv*, abs/2310.06762, 2023b.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual learning for large language models: A survey. *ArXiv*, abs/2402.01364, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Ke-Yang Chen, Kexin Yang, Mei Li, Min Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yunyang Wan, Yunfei Chu, Zeyu Cui, Zhenru Zhang, and Zhi-Wei Fan. Qwen2 technical report. *ArXiv*, abs/2407.10671, 2024.

Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. Scalable and order-robust continual learning with additive parameter decomposition. *arXiv: Learning*, 2019.

A Appendix

APPENDIX TEXT