

HarmoLoRA: Commit-Stage Update Admission in Low-Rank Adaptation for Continual Speech Recognition

Ye-Eun Ko Ji-Eun Choi Dong-Hyun Kim Joon-Hyuk Chang[†]

Hanyang University, Seoul, Republic of Korea

{yee772, uni1018, dlstjd1477, jchang}@hanyang.ac.kr

Abstract

Replay-free continual automatic speech recognition (ASR) requires adapting to new speech domains while maintaining a single persistent model without old data, task identifiers, or inference-time routing. We identify commit-stage update admission as a distinct post-training problem for continual low-rank adaptation (LoRA): after a task-wise LoRA update has been trained, the system must decide which part of that update should become persistent in the deployed model. This decision is especially important in ASR, where domains share phonetic and transcription structure but still differ in speakers, channels, noise, speaking style, and lexical coverage. We propose HarmoLoRA, a rank-constrained commit operator that decomposes the trained current-task update with respect to previously committed directions, constructs a rank-closed residual commit path, and uses a lexicographic constrained rule to choose the smallest residual admission coefficient among near-best realized candidates. The selected update is committed to one merged ASR checkpoint, requiring no replay, task identifiers, routing, gating, or multi-branch selection at inference. Across the reported ASR orders, HarmoLoRA improves recognition performance and reduces forgetting relative to reproduced replay-free and LoRA-based continual adaptation baselines.

1 Introduction

Automatic speech recognition (ASR) systems deployed in real environments must adapt to changing speakers, acoustic conditions, channels, speaking styles, and domain-specific vocabulary. Modern ASR benefits from deep acoustic modeling, noise robust recognition, and self-supervised or weakly supervised pretraining (Hinton et al., 2012; Li et al., 2014; Baevski et al., 2020; Hsu et al., 2021; Radford et al., 2023), but deployments rarely follow

a fixed distribution. Sequential adaptation to new domains can improve current domain recognition while overwriting behavior that was useful for earlier domains, a failure mode commonly described as catastrophic forgetting (McCloskey and Cohen, 1989; Parisi et al., 2019; Sadhu and Hermansky, 2020; Chang et al., 2021; Yang et al., 2022; Vander Eeck and Van hamme, 2023a).

We study replay-free continual ASR under a single-model deployment constraint. The model receives a stream of speech domains, accesses only current-task data during adaptation, and must remain one merged ASR model at inference time. Previous task training data are not stored or replayed. No task or domain identifier is available at test time, and the system cannot select task-specific adapters, route examples through domain-specific branches, or use multiple inference paths. This excludes episodic memory and task-conditioned masking assumptions used in some continual learning methods (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2019; Serra et al., 2018).

Unlike class incremental tasks with disjoint label spaces, continual ASR domains share phonetic structure, a transcription objective, and a decoder side subword space. TED talks, audiobooks, noisy speech, podcasts, and web video recordings differ in channel, acoustic conditions, speakers, style, and lexical coverage, but they are not independent recognition problems. Therefore, previously committed update directions can remain useful for new domains, and the central question is not how to isolate every update, but how much residual information should be admitted into the persistent model. This motivates HarmoLoRA’s separation between historical span reuse and residual admission at the commit stage.

Low-rank adaptation (LoRA) is attractive under this constraint because it freezes pretrained weights and learns low-rank matrices whose product forms

[†]Corresponding author.

a task-specific update (Hu et al., 2022). LoRA has also been used for Whisper-based multilingual ASR and parameter-efficient speaker-personalized ASR adaptation (Song et al., 2024; Joseph and Baby, 2024). Since a LoRA update can be merged into the backbone, LoRA supports single-model inference. However, committing every raw task update assumes that all trained updates are compatible, an assumption that is brittle in replay-free continual ASR.

Existing continual LoRA methods mainly control how current-task modules are trained, regularized, or composed. O-LoRA encourages orthogonal LoRA subspaces (Wang et al., 2023), InLoRA designs interference-aware LoRA adaptation (Liang and Li, 2024), and adapter-based continual ASR has explored small task-specific modules (Vander Eeck and Van hamme, 2023b). These approaches leave a complementary post-training problem. We identify *commit-stage update admission* as a distinct problem: after a LoRA update for a task has been trained, the system must decide which part becomes persistent in the deployed model.

HarmoLoRA addresses this problem as a post-training, rank-constrained commit operator. It decomposes the trained effective update into a historical span component, namely the current update component aligned with previously committed updates, and a residual component outside that span. HarmoLoRA scales the residual component and projects each candidate back to the original LoRA rank class before validation, so the commit decision is made over realized rank-constrained candidate updates rather than idealized full-rank update vectors.

HarmoLoRA selects the residual coefficient by a lexicographic constrained rule: it first restricts candidates to those whose realized current-task merge validation loss remains near the best candidate loss, and then chooses the candidate with the smallest residual admission coefficient. The selected candidate is not necessarily the absolute minimum loss candidate, and it is not chosen by geometric proximity to the raw update. Because previous task data are unavailable, HarmoLoRA uses merge validation only as a current-task plasticity guard and does not claim a global low loss guarantee for previous tasks.

The main contributions of this study are as follows:

- We formulate commit-stage update admission as a distinct post-training problem for replay-free continual ASR under a single persistent model constraint.
- We introduce HarmoLoRA, which decomposes a trained LoRA update, constructs a rank-closed residual commit path, realizes each candidate within the original LoRA rank class, and selects the smallest residual admission coefficient among near-best current-task candidates.
- We provide empirical and commit-stage diagnostics using selected residual coefficients and rank-constrained realization error to examine how historical span reuse and residual admission affect the persistent ASR model.

2 Background

2.1 Replay-Free Continual ASR

We consider a sequence of ASR tasks

$$\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_T, \quad (1)$$

where each task $\mathcal{D}_t$ consists of speech-transcript pairs

$$\mathcal{D}_t = \{(x_i^t, y_i^t)\}_{i=1}^{N_t}. \quad (2)$$

At task t , the model can access only the current-task data. Previous task training data are not replayed. At inference time, no task identifier or domain identifier is provided, and a single ASR model serves all tasks observed so far. This setting requires balancing plasticity for the current domain and stability for previously learned recognition behavior, while also accounting for shared ASR structure across domains. The same tension motivates regularization-based continual learning methods such as EWC and synaptic intelligence (Kirkpatrick et al., 2017; Zenke et al., 2017).

2.2 LoRA Update Merging

LoRA is one form of parameter-efficient finetuning, alongside adapter and bias-only tuning methods (Houlsby et al., 2019; Ben Zaken et al., 2022). It adapts a pretrained model by learning low-rank updates for selected linear layers while keeping the backbone weights frozen. Let

$$W_{\text{init}}^\ell \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}} \quad (3)$$

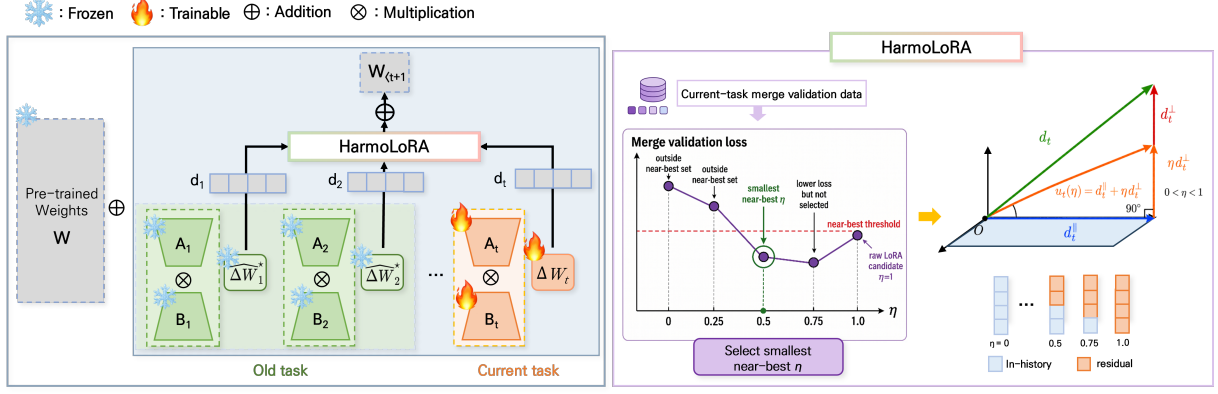


Figure 1: Overview of HarmoLoRA. After the LoRA update for a task is trained, it is projected onto the historical span, the residual path is realized under the original LoRA rank constraint, the resulting candidates are evaluated using merge validation, and the selected update is committed to the single persistent ASR model for subsequent tasks.

denote the initial weight of layer ℓ . For task t , LoRA learns

$$A_t^\ell \in \mathbb{R}^{d_{\text{out}} \times r}, \quad B_t^\ell \in \mathbb{R}^{r \times d_{\text{in}}}, \quad (4)$$

where $r \ll \min(d_{\text{out}}, d_{\text{in}})$. Throughout this paper, we use the convention

$$\Delta W_t^\ell = s_\ell A_t^\ell B_t^\ell, \quad (5)$$

where $s_\ell > 0$ is the LoRA scaling factor and ΔW_t^ℓ has the same shape as W_{init}^ℓ .

A direct sequential LoRA merge accumulates raw task updates:

$$W_{\text{direct}}^\ell = W_{\text{init}}^\ell + \sum_{i=1}^T \Delta W_i^\ell. \quad (6)$$

This produces a single inference model, but it treats all raw task updates as compatible. In replay-free continual ASR, this can be problematic because a current-task update may introduce domain-specific changes that interfere with previously useful recognition behavior.

2.3 Model Merging and Commit-Stage Selection

Model merging combines adapted models or task-specific updates into a single parameter set without retraining from scratch. Task-vector arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), and importance-guided or prediction-guided merging (Matena and Raffel, 2022; Jin et al., 2023) show that post-training structure can reduce interference. Perturb-and-Merge connects merging with

continual learning by combining previous and current task-specific models after each task (Qiu et al., 2025).

Weight space interpolation and averaging also motivate validation-guided merge selection. Mode connectivity, stochastic weight averaging, Model Soups, WiSE-FT, and LM-Cocktail show that averaging or interpolation can preserve single-model inference while improving robustness or generalization (Garipov et al., 2018; Izmailov et al., 2018; Wortsman et al., 2022a,b; Xiao et al., 2024).

HarmoLoRA differs from training stage continual LoRA methods and full model convex merging. It operates after LoRA training for each task, decomposes the trained update into historical span and residual components, and selects a rank-constrained residual admission path for one persistent ASR model. Without previous task data, validation selection only guards current-task plasticity.

3 Methodology

We propose HarmoLoRA, a rank-constrained commit-stage update admission operator for replay-free continual ASR. Figure 1 illustrates the overall commit-stage procedure of HarmoLoRA. For each task, HarmoLoRA trains a standard LoRA update using only current-task data and then decides how much of that update should be committed to the persistent single model. In this paper, *commit* denotes accepting a selected update into the persistent model, whereas *merge* denotes materializing an update by adding it to the backbone weights for validation or inference. HarmoLoRA is therefore a

post-training operator, not a training regularizer.

3.1 LoRA Update for Each Task and Historical Projection

Let $\mathcal{I}$ denote the set of layers to which LoRA is applied. Before task t , the persistent model contains the selected updates committed from previous tasks. The persistent weight for layer ℓ is defined as follows:

$$W_{<t}^\ell = W_{\text{init}}^\ell + \sum_{i=1}^{t-1} \widehat{\Delta W}_i^{\ell,*}, \quad \ell \in \mathcal{I}, \quad (7)$$

where W_{init}^ℓ denotes the initial pretrained weight of layer ℓ , and $\widehat{\Delta W}_i^{\ell,*}$ denotes the selected update committed after task i . When the layer superscript is omitted, we use $\widehat{\Delta W}_i^* = \{\widehat{\Delta W}_i^{\ell,*}\}_{\ell \in \mathcal{I}}$ to denote the collection of committed updates over all LoRA-applied layers. For the first task, no previous update exists, and we set $W_{<1}^\ell = W_{\text{init}}^\ell$. At task t , we freeze $W_{<t}$, attach a fresh LoRA module, and optimize only the LoRA parameters on the current task. The forward transformation at layer ℓ is defined as follows:

$$h_t^\ell = W_{<t}^\ell x + \Delta W_t^\ell x, \quad \Delta W_t^\ell = s_\ell A_t^\ell B_t^\ell, \quad (8)$$

where h_t^ℓ denotes the hidden activation at layer ℓ , x denotes the layer input, A_t^ℓ and B_t^ℓ are the task- t LoRA matrices, and $s_\ell > 0$ denotes the LoRA scaling factor. The current-task LoRA module is trained with the ordinary ASR objective on $\mathcal{D}_t^{\text{train}}$. HarmoLoRA does not use replay, distillation, orthogonality, or other continual learning losses during this training stage. After training, the raw effective LoRA update over all LoRA-applied layers is denoted as follows:

$$\Delta W_t = \{\Delta W_t^\ell\}_{\ell \in \mathcal{I}}. \quad (9)$$

We vectorize this update over the LoRA-applied layers as follows:

$$d_t = \text{vec}_{\mathcal{I}}(\Delta W_t) \in \mathbb{R}^D, \quad D = \sum_{\ell \in \mathcal{I}} d_{\text{out}}^\ell d_{\text{in}}^\ell, \quad (10)$$

where D denotes the effective update dimension over LoRA-applied layers, rather than the full ASR model parameter dimension. The operator $\text{vec}_{\mathcal{I}}(\cdot)$ vectorizes the updates of the LoRA-applied layers using a fixed vectorization convention and concatenates them in a fixed layer order to form a single

vector. For each previous task $i < t$, the vectorized committed update is denoted as follows:

$$u_i^* = \text{vec}_{\mathcal{I}}(\widehat{\Delta W}_i^*). \quad (11)$$

Using these committed update vectors, the historical update span is defined as follows:

$$\mathcal{S}_{t-1} = \text{span}(u_1^*, \dots, u_{t-1}^*). \quad (12)$$

For $t > 1$, the current update is decomposed into its historical span component and residual component as follows:

$$d_t^\parallel = P_{\mathcal{S}_{t-1}} d_t, \quad d_t^\perp = d_t - d_t^\parallel, \quad (13)$$

where $P_{\mathcal{S}_{t-1}}$ denotes the orthogonal projection onto the historical update span. The term $d_t^\parallel$ denotes the component of the newly trained current-task update that lies in the span of previously committed directions, whereas $d_t^\perp$ denotes the residual outside that span. Although $d_t^\parallel$ lies in the span of previous updates, it is not old knowledge added a second time; it is the projection of the current-task update.

In implementation, HarmoLoRA does not explicitly construct an orthonormal basis for $\mathcal{S}_{t-1}$. Let the matrix of previous committed update vectors be denoted as follows:

$$U_{t-1} = [u_1^*, \dots, u_{t-1}^*]. \quad (14)$$

For $t > 1$, since the columns of U_{t-1} span $\mathcal{S}_{t-1}$, the orthogonal projection operator is explicitly defined as follows:

$$P_{\mathcal{S}_{t-1}} = U_{t-1} (U_{t-1}^\top U_{t-1})^\dagger U_{t-1}^\top, \quad (15)$$

where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse. The projection coefficients are computed as follows:

$$\begin{aligned} G_{t-1} &= U_{t-1}^\top U_{t-1}, \\ b_t &= U_{t-1}^\top d_t, \\ c_t &= G_{t-1}^\dagger b_t, \end{aligned} \quad (16)$$

where G_{t-1} denotes the Gram matrix of previous committed updates, b_t denotes the vector of inner products between the current update and the previous committed updates, and $G_{t-1}^\dagger$ denotes the Moore-Penrose pseudoinverse. The historical span component is then computed as follows:

$$d_t^\parallel = U_{t-1} c_t. \quad (17)$$

All inner products are computed by summing Frobenius inner products across the LoRA-applied layers. For the first task, no historical span exists, so the raw LoRA update is committed.

3.2 Rank-Closed Residual Commit Path

For $t > 1$, HarmoLoRA constructs a residual commit path by scaling the residual component of the current update. The candidate update vector is defined as follows:

$$u_t(\eta) = d_t^\parallel + \eta d_t^\perp, \quad \eta \in [0, 1], \quad (18)$$

where η denotes the residual admission coefficient that controls how much of the residual component $d_t^\perp$ is admitted into the committed update. The discrete candidate set is defined as follows:

$$\mathcal{E} = \{0, 0.125, 0.25, 0.5, 0.75, 1.0\}. \quad (19)$$

This residual path connects the historical span component to the raw LoRA update, as follows:

$$u_t(0) = d_t^\parallel, \quad u_t(1) = d_t. \quad (20)$$

For each residual coefficient η , we recover the corresponding effective update for each layer. Here, $\text{mat}_{\mathcal{I}}(\cdot)$ denotes the inverse of $\text{vec}_{\mathcal{I}}(\cdot)$, restoring the vectorized update to the corresponding collection of layer-wise matrices under the same fixed layer order and vectorization convention:

$$M_t(\eta) = \text{mat}_{\mathcal{I}}(u_t(\eta)). \quad (21)$$

Accordingly, $M_t(\eta) = \{M_t^\ell(\eta)\}_{\ell \in \mathcal{I}}$. Unlike unconstrained task-vector interpolation, HarmoLoRA does not select in a full-rank update space. Although the raw LoRA update is rank- r per layer, the projected and residual-mixed candidate $M_t^\ell(\eta)$ may not remain rank- r . Each candidate is therefore projected back to the original LoRA rank class before validation. The realized candidate is defined as follows:

$$\widehat{\Delta W}_t^\ell(\eta) = \Pi_r \left(M_t^\ell(\eta) \right), \quad (22)$$

where the rank- r projection operator is defined as follows:

$$\Pi_r(X) = \arg \min_{\text{rank}(Z) \leq r} \|X - Z\|_F^2. \quad (23)$$

In practice, Π_r is implemented by applying truncated SVD to each layer. This step is not an additional selection criterion; it keeps each candidate within the LoRA update class. The commit decision is therefore made over realized rank-constrained candidate updates, not idealized full-rank vectors. We call this construction a rank-closed residual commit path. For $\eta = 1$, before numerical SVD realization, $M_t^\ell(1)$ equals

the raw LoRA update. Since the raw LoRA update is already rank-constrained by LoRA, truncated SVD recovers it up to numerical precision. The rank constraint is applied at the task update level even though the deployed system is a single merged ASR model, keeping stored committed updates consistent for future projections. We use $\widehat{\Delta W}_t(\eta) = \{\widehat{\Delta W}_t^\ell(\eta)\}_{\ell \in \mathcal{I}}$ to denote the realized candidate update over all LoRA-applied layers.

3.3 Merge Validation Selection

HarmoLoRA selects the residual coefficient using the current-task merge validation split. Let $\mathcal{L}_t^{\text{merge}}$ denote the merge validation loss. For each candidate, the merge validation loss is defined as follows:

$$\mathcal{L}_t(\eta) = \mathcal{L}_t^{\text{merge}} \left(f(\cdot; W_{<t} + \widehat{\Delta W}_t(\eta)) \right), \quad (24)$$

where $f(\cdot; \cdot)$ denotes the ASR model after temporarily merging the candidate update. No back-propagation is performed during this step. Each candidate is evaluated as a temporary merged ASR model, making selection consistent with the final single-model inference setting. For merge validation selection, the best candidate loss and the zero residual candidate loss are denoted as follows:

$$\mathcal{L}_t^{\min} = \min_{\eta \in \mathcal{E}} \mathcal{L}_t(\eta), \quad \mathcal{L}_t^0 = \mathcal{L}_t(0). \quad (25)$$

The near-best candidate set is defined as follows:

$$\mathcal{N}_t(\tau) = \left\{ \eta \in \mathcal{E} : \mathcal{L}_t(\eta) \leq \mathcal{L}_t^{\min} + \tau [\mathcal{L}_t^0 - \mathcal{L}_t^{\min}]_+ \right\}, \quad (26)$$

where $[z]_+ = \max(z, 0)$, and $\tau \geq 0$ denotes a fixed tolerance parameter. Equivalently, HarmoLoRA solves the following lexicographic constrained selection problem:

$$\begin{aligned} \eta_t^* &= \arg \min_{\eta \in \mathcal{E}} \eta \\ \text{s.t. } & \mathcal{L}_t(\eta) \leq \mathcal{L}_t^{\min} + \tau [\mathcal{L}_t^0 - \mathcal{L}_t^{\min}]_+. \end{aligned} \quad (27)$$

This rule first restricts candidates to those whose realized merge validation loss is near the best candidate loss, and then chooses the smallest residual admission coefficient. Thus, the selected candidate is not necessarily the minimum loss candidate and is not chosen by geometric proximity to the raw update. The selected task update and its vectorized form are denoted as follows:

$$\widehat{\Delta W}_t^* = \widehat{\Delta W}_t(\eta_t^*), \quad u_t^* = \text{vec}_{\mathcal{I}} \left(\widehat{\Delta W}_t^* \right). \quad (28)$$

The vector u_t^* is stored and used as a historical committed direction for future projection.

3.4 Final Commit

After task t , HarmoLoRA updates the persistent model by committing the selected update as follows:

$$W_{<t+1}^\ell = W_{<t}^\ell + \widehat{\Delta W}_t^{\ell,*}. \quad (29)$$

After all tasks have been processed, the final inference weight is defined as follows:

$$W_{\text{final}}^\ell = W_{\text{init}}^\ell + \sum_{i=1}^T \widehat{\Delta W}_i^{\ell,*}. \quad (30)$$

The final model is a single merged ASR model.

4 Experiments

4.1 Datasets

Each speech domain was treated as one continual adaptation task. We used publicly available ASR corpora under their original licenses or terms for academic ASR research, collected no new data, performed no additional speaker de-identification, re-distributed no raw audio, transcripts, speaker meta-data, or utterance-level examples, and reported only aggregate WER results.

TED-LIUM2 (TED2). TED-LIUM2 contains TED talk speech with official development and test splits (Rousseau et al., 2014). We used TED2 as a presentation speech domain and reported WER on the held-out test split following our continual ASR protocol.

CHiME3. CHiME3 is a noisy ASR benchmark recorded in real acoustic environments (Barker et al., 2015). We used CHiME3 as a noisy speech domain and evaluated all methods on the held-out noisy speech test split.

LibriSpeech. LibriSpeech is an audiobook-based read speech corpus (Panayotov et al., 2015). We used the “other” condition, denoted as LibriSpeech-other, as a challenging read speech domain. The task data were split into disjoint adaptation and merge validation subsets according to our continual ASR protocol, and evaluation was performed on the held-out LibriSpeech-other evaluation split.

Common Voice (CV). Common Voice is a crowdsourced ASR corpus with heterogeneous speakers, recording devices, and acoustic conditions (Ardila et al., 2020). We partitioned the development split into disjoint adaptation and merge validation subsets and evaluated on a fixed 500-utterance test list.

Order	Task sequence
1	LibriSpeech-other $\rightarrow$ CHiME3 $\rightarrow$ TED2
2	TED2 $\rightarrow$ LibriSpeech-other $\rightarrow$ CHiME3
3	CHiME3 $\rightarrow$ LibriSpeech-other $\rightarrow$ TED2
4	TED2 $\rightarrow$ GigaSpeech-YOU $\rightarrow$ LibriSpeech-other $\rightarrow$ CV $\rightarrow$ GigaSpeech-POD $\rightarrow$ CHiME3

Table 1: Task orders used in the main continual ASR experiments.

GigaSpeech-POD and GigaSpeech-YOU. GigaSpeech is a large-scale multi-domain ASR corpus (Chen et al., 2021). We used its podcast and YouTube subsets to construct the longer six-domain continual stream. For these subsets, we used two hours of development data and three hours of held-out test data.

4.2 Evaluation Metrics

We report word error rate (WER), where lower values are better:

$$\text{WER} = \frac{S + D + I}{N} \times 100, \quad (31)$$

where S , D , and I are substitutions, deletions, and insertions, and N is the number of reference words. Let $\text{WER}_{i,t}$ denote the WER on evaluation task i after the model has been committed through task t . Last is the WER of the final model state, averaged over the fixed evaluation suite, and Avg is the trajectory level average of the same fixed suite WER over the continual adaptation process. WER space backward transfer (BWT) is

$$\text{BWT}_{\text{WER}} = \frac{1}{T-1} \sum_{i=1}^{T-1} \left(\min_{t \in \{i, \dots, T-1\}} \text{WER}_{i,t} - \text{WER}_{i,T} \right). \quad (32)$$

Positive values indicate that the final model improves over the best pre-final WER on previous tasks; negative values indicate degradation. Non-negative forgetting in WER space is

$$\text{Forgetting}_{\text{WER}} = \frac{1}{T-1} \sum_{i=1}^{T-1} \left[\text{WER}_{i,T} - \min_{t \in \{i, \dots, T-1\}} \text{WER}_{i,t} \right]_+, \quad (33)$$

where $[z]_+ = \max(z, 0)$. Lower forgetting is better.

Method	Order 1		Order 2		Order 3	
	Last ($\downarrow$)	Avg ($\downarrow$)	Last ($\downarrow$)	Avg ($\downarrow$)	Last ($\downarrow$)	Avg ($\downarrow$)
EWC (Kirkpatrick et al., 2017) [†]	14.70 $\pm$ 0.25	18.21 $\pm$ 0.23	27.23 $\pm$ 0.51	19.28 $\pm$ 0.37	13.20 $\pm$ 0.31	21.08 $\pm$ 0.28
IncLoRA (Wang et al., 2023) [†]	13.80 $\pm$ 0.18	17.83 $\pm$ 0.11	22.43 $\pm$ 0.29	18.33 $\pm$ 0.21	15.07 $\pm$ 0.23	18.92 $\pm$ 0.18
L2P (Wang et al., 2022) [†]	14.17 $\pm$ 0.08	16.10 $\pm$ 0.12	13.57 $\pm$ 0.18	15.14 $\pm$ 0.15	17.87 $\pm$ 0.21	20.79 $\pm$ 0.14
O-LoRA (Wang et al., 2023) [†]	14.27 $\pm$ 0.42	16.07 $\pm$ 0.34	14.33 $\pm$ 0.37	15.64 $\pm$ 0.29	15.93 $\pm$ 0.41	19.07 $\pm$ 0.32
InfLoRA (Liang and Li, 2024) [†]	13.60 $\pm$ 0.12	15.90 $\pm$ 0.08	14.63 $\pm$ 0.21	15.08 $\pm$ 0.09	15.30 $\pm$ 0.18	17.58 $\pm$ 0.11
HarmoLoRA	12.50 $\pm$ 0.17	15.56 $\pm$ 0.11	12.77 $\pm$ 0.20	14.60 $\pm$ 0.15	12.97 $\pm$ 0.25	17.39 $\pm$ 0.13

[†] Reproduced experiments.

Table 2: WER (%) results for Orders 1–3. We report mean and standard deviation over three trials.

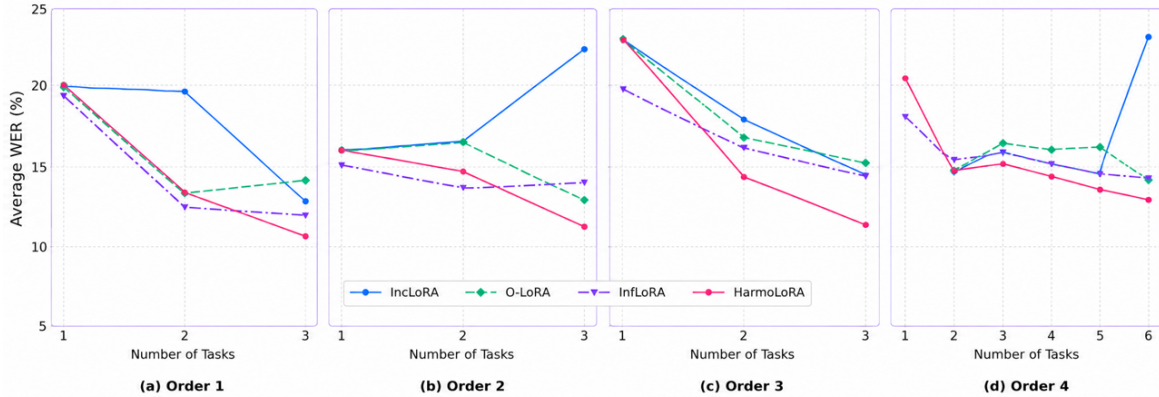


Figure 2: WER trajectories by task for IncLoRA, O-LoRA, InfLoRA, and HarmoLoRA.

4.3 Experimental Setup

We used Whisper small (Radford et al., 2023) as the ASR backbone and implemented all experiments in ESPnet (Watanabe et al., 2018). The backbone was frozen, and LoRA was applied to attention projection modules with rank $r = 4$, scaling $\alpha_{\text{LoRA}} = 8$, and dropout 0.05. The resulting ESPnet ASR model contained 241.47M total parameters, of which 884.74K parameters were trainable through LoRA. We used warmupLR with 1,500 warmup updates, a peak learning rate of 1.0×10^{-4} , and a single NVIDIA RTX A6000 GPU. Training took approximately 0.5 hours per task. All methods started from the same previously fine-tuned ASR checkpoint. At each task t , a newly initialized LoRA branch was trained only on the current adaptation split. Methods with commit candidates materialized each candidate into the backbone and selected it using the task-specific merge validation split. All methods used the same data partitions and decoding setup. HarmoLoRA, IncLoRA, O-LoRA, and InfLoRA were evaluated under the same single-model deployment constraint. L2P performs task-agnostic prompt selection without task identifiers, but it is reported only as a task-agnostic reference and is not directly comparable under the single-

model constraint.

For HarmoLoRA, after training the current-task LoRA update, we constructed rank-constrained residual commit candidates over the admission grid $\mathcal{E} = \{0, 0.125, 0.25, 0.5, 0.75, 1.0\}$. Each candidate was realized under the original LoRA rank constraint before merge validation, matching the rank-closed residual commit path. The near-best tolerance was fixed to $\tau = 0.1$ before the main comparison and was not tuned separately for each order. For Orders 1–3, we ran three random seeds and reported the mean and standard deviation. This fixed protocol kept the commit selection procedure consistent across orders.

5 Results

5.1 Main Continual ASR Results

Table 2 reports the main WER results across the three-domain orders. HarmoLoRA achieved the lowest Last WER in all three orders, with 12.50, 12.77, and 12.97 WER for Orders 1–3. Relative to the strongest reported non-HarmoLoRA result in each order, these corresponded to absolute gains of 1.10, 0.80, and 0.23 WER points. HarmoLoRA also obtained the lowest trajectory level Avg WER

Method	BWT ($\uparrow$)	Forgetting ($\downarrow$)
EWC (Kirkpatrick et al., 2017) [†]	-5.57	6.17
IncLoRA (Wang et al., 2023) [†]	-2.75	4.43
L2P (Wang et al., 2022) [†]	-0.38	0.90
O-LoRA (Wang et al., 2023) [†]	-2.02	2.38
InfLoRA (Liang and Li, 2024) [†]	-1.35	1.35
HarmoLoRA	0.08	0.65

Table 3: Average BWT and forgetting across the three-domain task sequences in Orders 1–3. Higher BWT and lower forgetting indicate better retention of performance on previously learned domains.

Method	Last ($\downarrow$)	Avg ($\downarrow$)	BWT ($\uparrow$)	Forgetting ($\downarrow$)
EWC	18.85	17.76	-3.08	3.38
IncLoRA	23.78	17.74	-11.84	11.84
L2P	14.98	15.82	-2.22	2.22
O-LoRA	14.68	16.63	-1.04	1.36
InfLoRA	14.85	15.98	-2.24	2.24
HarmoLoRA	13.68	15.63	-1.04	1.18

Table 4: Order 4 six-domain results with WER, BWT, and forgetting metrics.

in all three orders, with 15.56, 14.60, and 17.39 WER, showing that the improvement was not limited to the final model state.

The baseline comparison highlights stability–plasticity behavior. IncLoRA showed large order-dependent degradation, especially in Order 2, indicating that directly committing raw task updates can be unstable under domain order changes. O-LoRA and InfLoRA were stronger LoRA-based baselines, but HarmoLoRA improved both Last and Avg WER across all three orders, suggesting that the commit-stage admission decision matters even when LoRA training for each task is unchanged. HarmoLoRA achieved the highest BWT, 0.08, and the lowest forgetting, 0.65, in Table 3.

Figure 2 shows the fixed suite WER after each commit. The trajectories make two aspects of the table results explicit. First, the final advantage of HarmoLoRA is not produced by one isolated order: its terminal trajectory is the lowest in each of Orders 1–3 and remains the lowest in the longer Order 4 diagnostic. Second, the strongest baselines can be competitive at an intermediate model state but subsequently lose that advantage after another raw or constrained update is committed. HarmoLoRA instead makes the admission decision on the realized merged model, which is the model that will be carried to the next task.

Order	Task1	Task2	Task3	Task4	Task5	Task6
1	1.0	0.75	0.25	–	–	–
2	1.0	0.25	0.50	–	–	–
3	1.0	0.125	0.25	–	–	–
4	1.0	0.25	0.125	0.125	0.0	1.0

Table 5: Selected HarmoLoRA residual coefficients η_t^* . For Task 1, $\eta = 1.0$ is assigned by convention because no historical span exists and the raw LoRA update is committed.

5.2 Longer Domain Stream

Table 4 reports the longer six-domain stream consisting of TED2, GigaSpeech-YOU, LibriSpeech-other, CV, GigaSpeech-POD, and CHiME3. This experiment is used as a long stream diagnostic because it differs from Orders 1–3 in both stream length and domain composition. HarmoLoRA achieved 13.68 Last WER and 15.63 Avg WER, giving the best performance on both metrics. Compared with O-LoRA, HarmoLoRA reduced Last and Avg WER by 1.00 point each. It also tied O-LoRA for the best BWT (−1.04), while obtaining lower forgetting.

Compared with IncLoRA, HarmoLoRA reduced Last WER by 10.10 points, Avg WER by 2.11 points, and forgetting by 10.66 points. These results provide auxiliary evidence that commit-stage residual admission can remain useful when the persistent model must absorb a longer sequence of heterogeneous speech domain shifts.

5.3 Analysis of Selected Residual Coefficients

Table 5 shows that the selected residual coefficient changes with both the current domain and its position in the stream. The first task receives $\eta = 1.0$ by convention because no historical span is available. From the second task onward, HarmoLoRA selects coefficients from the same fixed candidate set according to the merge validation loss of the realized candidate model.

The same domain can receive different coefficients under different historical spans. CHiME3 is assigned $\eta = 0.75$ in Order 1, $\eta = 0.50$ in Order 2, and $\eta = 1.0$ at the end of Order 4. LibriSpeech-other is assigned $\eta = 0.25$ after TED2 in Order 2, but $\eta = 0.125$ in Orders 3 and 4. TED2 receives $\eta = 0.25$ when it appears after previous domains in Orders 1 and 3. These changes show that the commit decision is not a fixed coefficient associated with a dataset; it depends on the current trained update, the span accumulated before that task, and

the losses of the realized candidates.

Order 4 also includes both endpoints of the residual path. GigaSpeech-POD selects $\eta = 0.0$, so its committed update is obtained from the historical span component without an additional residual term. The final CHiME3 task selects $\eta = 1.0$, which admits the full residual component of the trained update before rank-constrained realization. Thus, the selected coefficients cover the full candidate range rather than following a uniform shrinkage pattern.

5.4 Trajectory Level Interpretation

Last, Avg, BWT, and forgetting summarize complementary aspects of the continual adaptation trajectory. Last measures only the final merged model state, whereas Avg also reflects the intermediate model states after successive commits. A method can therefore recover at the final task while still obtaining a worse Avg if its intermediate model states are unstable. HarmoLoRA achieves the lowest Last and Avg WER in all three main orders, indicating that its advantage is observed both at the final model state and across the adaptation trajectory. In Order 4, HarmoLoRA obtains a BWT of -1.04 , indicating that the final model state is, on average, worse than the best pre-final reference on previous tasks. At the same time, its forgetting is 1.18 , the lowest among the evaluated methods in the same stream. Thus, HarmoLoRA reduces previous task degradation in the longer heterogeneous stream, although it does not eliminate it. Because residual admission is selected using only current-task merge validation, retention remains an empirical outcome rather than an explicit constraint in the selection rule.

6 Conclusion

This study proposed HarmoLoRA, a commit-stage update admission operator for replay-free continual ASR with low-rank adaptation. HarmoLoRA decomposed each LoRA update into historical span and residual components, constructed a rank-closed residual commit path, and selected the smallest residual admission coefficient whose realized merged model preserved near-best current-task validation loss. The selected update yielded a single persistent ASR model without replay or inference-time routing. Experiments showed that HarmoLoRA improved the stability-plasticity profile within the evaluated ASR streams. Across the

evaluated domain orders, it achieved the lowest Last and Avg WER, and the best or tied-best retention results among the compared baselines. HarmoLoRA committed the historical span component of each trained update and admitted only as much of the residual component as was needed to preserve near-best merge validation loss in the single merged ASR model.

Limitations

The claims of this paper are scoped to replay-free continual ASR under a single-model deployment constraint. HarmoLoRA is a post-training commit operator that uses only the current-task merge validation split to select residual admission coefficients. Because no previous task training data are used for selection, previous task preservation should be interpreted as an empirical preservation outcome measured on held-out evaluation sets, rather than as a property guaranteed by the selection rule. The positive WER space BWT and low forgetting reported in the main experiments therefore support the claim within the evaluated ASR domain streams, but they should not be read as a claim of monotonic protection for every previous task or domain.

The evaluation focused on Whisper small with LoRA applied to attention projection modules, and the main residual coefficient grid was fixed to $\{0, 0.125, 0.25, 0.5, 0.75, 1.0\}$ with $\tau = 0.1$. Larger ASR backbones, different adapter placements, different LoRA ranks, and multilingual or streaming ASR settings may exhibit different update geometry. Order 4 is reported as a long stream diagnostic rather than main comparative evidence. Broader long stream evaluations are needed to assess robustness over longer task sequences.

Whisper was pretrained on a large web-sourced corpus whose complete itemized composition is not publicly available. Prior exposure to public corpora used in our evaluation therefore cannot be excluded.

By not storing or replaying previous task speech-transcript training examples, HarmoLoRA reduces reliance on retained previous task data. Nevertheless, ASR deployment can involve sensitive speech recordings, and transcription errors may affect downstream applications. These considerations are broader than HarmoLoRA itself and should be accounted for when deploying continual ASR systems.

Acknowledgments

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00557944) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program (Hanyang University)).

ChatGPT (OpenAI) was used only for grammar correction and clarity refinement. The authors reviewed and are fully responsible for all content.

References

- Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. [Common voice: A massively-multilingual speech corpus](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.
- Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. [wav2vec 2.0: A framework for self-supervised learning of speech representations](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460.
- Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe. 2015. [The third “CHiME” speech separation and recognition challenge: Dataset, task and baselines](#). In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 504–511.
- Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. [BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.
- Heng-Jui Chang, Hung-yi Lee, and Lin-shan Lee. 2021. [Towards lifelong learning of end-to-end ASR](#). In *Interspeech 2021*, pages 2551–2555.
- Arslan Chaudhry, Marc’Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. [Efficient lifelong learning with A-GEM](#). In *International Conference on Learning Representations*.
- Cheng Chen, Pengpeng Zeng, Yuyu Guo, Lianli Gao, Hengtao Shen, and Jingkuan Song. 2026. [JANUS-LoRA: A balanced low-rank adaptation for continual learning](#). In *International Conference on Machine Learning*.
- Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. [GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio](#). In *Interspeech 2021*, pages 3670–3674.
- Carl Eckart and Gale Young. 1936. [The approximation of one matrix by another of lower rank](#). *Psychometrika*, 1:211–218.
- Timur Garipov, Pavel Izmailov, Dmitrii Podoprikin, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. [Loss surfaces, mode connectivity, and fast ensembling of DNNs](#). In *Advances in Neural Information Processing Systems*, volume 31.
- Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. 2012. [Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups](#). *IEEE Signal Processing Magazine*, 29(6):82–97.
- 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.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhota, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. [HuBERT: Self-supervised speech representation learning by masked prediction of hidden units](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hananeh Hajishirzi, and Ali Farhadi. 2023. [Editing models with task arithmetic](#). In *International Conference on Learning Representations*.
- Pavel Izmailov, Dmitrii Podoprikin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. [Averaging weights leads to wider optima and better generalization](#). In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
- Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. [Dataless knowledge fusion by merging weights of language models](#). In *International Conference on Learning Representations*.

- George Joseph and Arun Baby. 2024. [Speaker personalization for automatic speech recognition using weight-decomposed low-rank adaptation](#). In *Interspeech 2024*, pages 2875–2879.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. [Overcoming catastrophic forgetting in neural networks](#). *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.
- Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. 2014. [An overview of noise-robust automatic speech recognition](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):745–777.
- Yan-Shuo Liang, Jia-Rui Chen, and Wu-Jun Li. 2025. [Gated integration of low-rank adaptation for continual learning of large language models](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 76577–76607.
- Yan-Shuo Liang and Wu-Jun Li. 2024. [InfLoRA: Interference-free low-rank adaptation for continual learning](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647.
- David Lopez-Paz and Marc’Aurelio Ranzato. 2017. [Gradient episodic memory for continual learning](#). In *Advances in Neural Information Processing Systems*, volume 30.
- Michael S. Matena and Colin A. Raffel. 2022. [Merging models with fisher-weighted averaging](#). In *Advances in Neural Information Processing Systems*, volume 35.
- Michael McCloskey and Neal J. Cohen. 1989. [Catastrophic interference in connectionist networks: The sequential learning problem](#). In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press.
- Leon Mirsky. 1960. [Symmetric gauge functions and unitarily invariant norms](#). *The Quarterly Journal of Mathematics*, 11(1):50–59.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. [LibriSpeech: An ASR corpus based on public domain audio books](#). In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5206–5210.
- German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. [Continual lifelong learning with neural networks: A review](#). *Neural Networks*, 113:54–71.
- Haomiao Qiu, Miao Zhang, Ziyue Qiao, and Liqiang Nie. 2025. [Train with perturbation, infer after merging: A two-stage framework for continual learning](#). In *Advances in Neural Information Processing Systems*, volume 38.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. [Robust speech recognition via large-scale weak supervision](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.
- Anthony Rousseau, Paul Deléglise, and Yannick Estève. 2014. [Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 3935–3939, Reykjavik, Iceland. European Language Resources Association.
- Samik Sadhu and Hynek Hermansky. 2020. [Continual learning in automatic speech recognition](#). In *Interspeech 2020*, pages 1246–1250.
- Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. [Overcoming catastrophic forgetting with hard attention to the task](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4548–4557. PMLR.
- Zhesu Song, Jianheng Zhuo, Yifan Yang, Ziyang Ma, Shixiong Zhang, and Xie Chen. 2024. [LoRA-Whisper: Parameter-efficient and extensible multilingual ASR](#). In *Interspeech 2024*, pages 3934–3938.
- Steven Vander Eeckt and Hugo Van hamme. 2023a. [Rehearsal-free online continual learning for automatic speech recognition](#). In *Interspeech 2023*, pages 944–948.
- Steven Vander Eeckt and Hugo Van hamme. 2023b. [Using adapters to overcome catastrophic forgetting in end-to-end automatic speech recognition](#). In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5.
- Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. [Orthogonal subspace learning for language model continual learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, Singapore. Association for Computational Linguistics.
- Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. [Learning to prompt for continual learning](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149.
- Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. [ESPnet: End-to-end speech processing toolkit](#). In *Interspeech 2018*, pages 2207–2211.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022a. [Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2022b. [Robust fine-tuning of zero-shot models](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Xingrun Xing. 2024. [LM-cocktail: Resilient tuning of language models via model merging](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2474–2488, Bangkok, Thailand. Association for Computational Linguistics.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. [TIES-merging: Resolving interference when merging models](#). In *Advances in Neural Information Processing Systems*, volume 36.

Muqiao Yang, Ian Lane, and Shinji Watanabe. 2022. [On-line continual learning of end-to-end speech recognition models](#). In *Interspeech 2022*, pages 2668–2672.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. [Continual learning through synaptic intelligence](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR.

Xin Zhang, Liang Bai, Xian Yang, and Jiye Liang. 2025. [C-LoRA: Continual low-rank adaptation for pre-trained models](#). *arXiv preprint arXiv:2502.17920*.

A Additional Analyses

A.1 Diagnostic Residual Norm Cap Variant

We evaluated a diagnostic HarmoLoRA-Cap variant to examine whether the improvement of HarmoLoRA could be explained by committing smaller residual updates alone. This variant was not intended as a separate continual learning method. Instead, it served as a control experiment that preserved the main projection structure of HarmoLoRA while adding an explicit size limit to the new residual component before commitment.

The Cap variant used the same historical projection as HarmoLoRA. Given the current update d_t ,

it first decomposed the update into the historical span component and the residual component:

$$d_t^{\parallel} = P_{\mathcal{S}_{t-1}} d_t, \quad d_t^{\perp} = d_t - d_t^{\parallel}. \quad (34)$$

HarmoLoRA forms a residual path candidate as

$$u_t(\eta) = d_t^{\parallel} + \eta d_t^{\perp}. \quad (35)$$

The Cap variant modified only the residual term. For each residual coefficient η , it replaced $\eta d_t^{\perp}$ with a norm-capped residual:

$$\tilde{r}_t(\eta) = \gamma_t(\eta) \eta d_t^{\perp}, \quad (36)$$

where

$$\gamma_t(\eta) = \min \left(1, \frac{C_t}{\|\eta d_t^{\perp}\|_2 + \epsilon} \right). \quad (37)$$

Here C_t denotes the residual norm budget used by the diagnostic variant, and ϵ is a small constant for numerical stability. For each task, C_t was fixed before candidate evaluation and was not tuned using previous task evaluation data. The resulting candidate update is

$$u_t^{\text{cap}}(\eta) = d_t^{\parallel} + \tilde{r}_t(\eta). \quad (38)$$

Thus, the Cap variant kept the historical span term unchanged during candidate construction while restricting the norm of the newly introduced residual term. It did not use previous task data, introduce an additional training loss, or require inference-time routing. Its role was to test whether explicit residual size restriction alone could reproduce the behavior of HarmoLoRA.

Table 6 reports the diagnostic comparison. HarmoLoRA achieved lower Last WER than the Cap variant in all three orders and lower Avg WER in Orders 1 and 2. The Cap variant obtained a slightly lower Avg WER in Order 3, but its Last WER was worse than that of HarmoLoRA. This result indicates that residual size restriction alone did not consistently reproduce the final-model performance of HarmoLoRA.

The comparison separates two effects. The Cap variant directly constrained the magnitude of the residual component. HarmoLoRA, in contrast, evaluated realized rank-constrained candidates after temporary merging and selected the smallest residual coefficient whose current-task merge validation loss remained near the best candidate loss. The stronger final WER profile of HarmoLoRA suggests that the observed gains were not explained only by committing a smaller residual update.

Method	Order 1		Order 2		Order 3	
	Last ($\downarrow$)	Avg ($\downarrow$)	Last ($\downarrow$)	Avg ($\downarrow$)	Last ($\downarrow$)	Avg ($\downarrow$)
HarmoLoRA	12.50	15.56	12.77	14.60	12.97	17.39
HarmoLoRA-Cap (<i>diagnostic</i>)	14.13	15.87	12.83	14.76	13.17	17.34

Table 6: Diagnostic comparison between HarmoLoRA and the residual norm Cap variant. The Cap variant used the same historical projection as HarmoLoRA but explicitly limited the norm of the newly introduced residual term. It was used only to test whether residual size restriction alone could explain the behavior of HarmoLoRA.

Task	Domain	Selected η	Committed norm	Mean SVD rel. error	Max SVD rel. error
1	TED2	1.000	1.86541	0.0000	0.0000
2	LibriSpeech-other	0.250	0.84094	3.55×10^{-2}	7.38×10^{-2}
3	CHiME3	0.500	0.23055	1.04×10^{-3}	2.52×10^{-3}

Table 7: Rank-constrained SVD realization error of the selected HarmoLoRA updates in the Order 2 diagnostic stream, TED2 $\rightarrow$ LibriSpeech-other $\rightarrow$ CHiME3. The errors quantify the mismatch introduced when the selected effective update was returned to the original LoRA rank class.

Task	$\eta = 0$	$\eta = 0.125$	$\eta = 0.25$	$\eta = 0.5$	$\eta = 0.75$	$\eta = 1.0$ raw	Near-best threshold	Selected η
LibriSpeech-other	7.06022	7.03657	7.02833	7.04634	7.13913	7.27663	7.03152	0.25
CHiME3	8.32076	8.29510	8.27522	<u>8.25101</u>	8.24919	8.26955	8.25635	0.50

Table 8: Candidate merge validation losses along the residual merge path for the Order 2 diagnostic stream. Bold values indicate the minimum loss candidate. Underlined values indicate additional near-best candidates. HarmoLoRA selects the smallest near-best η , not necessarily the absolute minimum loss η .

A.2 Rank-Constrained Realization Error

Table 7 reports the error introduced when the selected candidate update was realized under the original LoRA rank constraint in the Order 2 diagnostic stream. We used truncated SVD because it gives the best rank- r approximation under standard unitarily invariant norms, including the Frobenius norm (Eckart and Young, 1936; Mirsky, 1960). Therefore, the reported SVD error was not an artifact of an arbitrary factorization method. It measured the minimum reconstruction mismatch induced by returning the selected candidate to the LoRA rank class.

There is no universal numerical threshold that makes a rank-constrained realization acceptable for every neural network layer or task. We therefore used two checks. First, we reported the relative reconstruction error with respect to the committed update norm. Second, candidates were evaluated after rank-constrained realization, rather than before it. Thus, if the SVD realization had substantially damaged a candidate update, the merge validation loss would have reflected that degradation.

The measured errors were small relative to the committed updates. Task 1 had zero error because

the raw LoRA update was already rank constrained. For LibriSpeech-other, the mean relative SVD error was 3.55×10^{-2} , with a maximum of 7.38×10^{-2} . For CHiME3, the mean and maximum relative errors were 1.04×10^{-3} and 2.52×10^{-3} , respectively. Together with the realized candidate validation step, these values indicate that the selected updates were not dominated by factorization mismatch.

A.3 Candidate Merge Validation Losses

Table 8 reports the candidate validation losses used to select residual coefficients in the Order 2 diagnostic stream. For LibriSpeech-other, the minimum validation loss was obtained at $\eta = 0.25$, which was also selected. For CHiME3, the minimum loss was obtained at $\eta = 0.75$, but $\eta = 0.5$ was within the near-best range and was selected by the conservative rule.

Figures 3 and 4 visualize the same candidate loss landscapes along the residual merge path. These figures make the selection behavior explicit. For LibriSpeech-other, the minimum loss candidate and the selected candidate coincide at $\eta = 0.25$. For CHiME3, however, the absolute minimum occurs at $\eta = 0.75$, while $\eta = 0.50$ remains within the

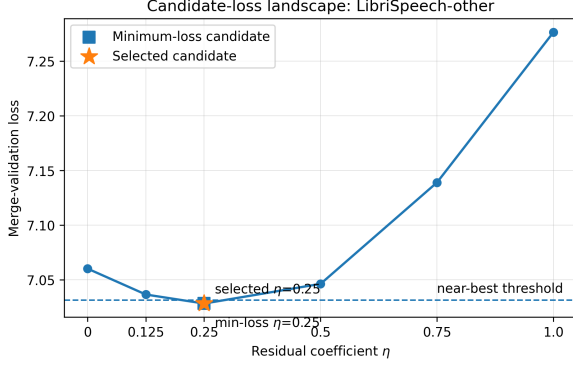


Figure 3: Candidate merge validation loss landscape for LibriSpeech-other in the Order 2 diagnostic stream. The selected residual coefficient is $\eta = 0.25$, which also achieves the minimum merge validation loss. The near-best threshold is shown to indicate the selection region used by HarMoLoRA.

near-best threshold and is selected as the smaller residual coefficient. This shows that HarMoLoRA is not a fixed shrinkage rule and does not always choose the raw update or the absolute minimum loss candidate. Instead, it selects the smallest residual coefficient that preserves near-best merge validation loss after rank-constrained realization.

Under the main selection rule with $\tau = 0.1$, the near-best threshold is

$$\mathcal{L}_t^{\min} + \tau [\mathcal{L}_t^0 - \mathcal{L}_t^{\min}]_+ . \quad (39)$$

For LibriSpeech-other, this gives

$$7.02833 + 0.1(7.06022 - 7.02833) = 7.03152, \quad (40)$$

so only $\eta = 0.25$ qualified. For CHiME3, the threshold was

$$8.24919 + 0.1(8.32076 - 8.24919) = 8.25635, \quad (41)$$

so both $\eta = 0.5$ and $\eta = 0.75$ qualified. HarMoLoRA selected the smaller coefficient, $\eta = 0.5$, because it preserved near-best validation loss while admitting a smaller residual component.

A.4 Interpretation of Residual Coefficients by Dataset and Order

Table 5 shows that the residual coefficient selected by HarMoLoRA depends on both the current domain and its position in the continual stream. This behavior is expected because the residual coefficient is selected from the realized merge validation loss, not from dataset identity alone. The same

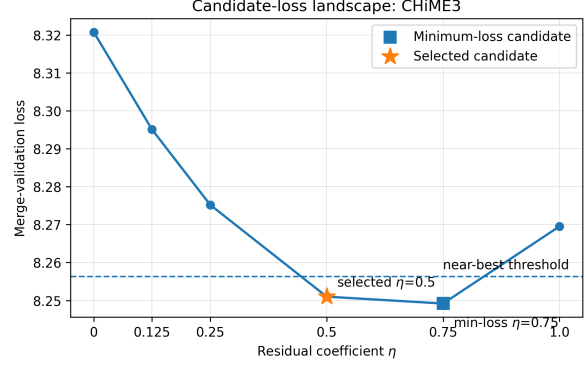


Figure 4: Candidate merge validation loss landscape for CHiME3 in the Order 2 diagnostic stream. The minimum loss is obtained at $\eta = 0.75$, but $\eta = 0.50$ remains within the near-best region and is selected by the conservative HarMoLoRA rule. This illustrates that HarMoLoRA does not simply choose the absolute minimum loss candidate or the full raw update.

dataset can therefore receive different coefficients depending on which directions have already been committed to the historical span.

For the first task in every order, $\eta = 1.0$ is assigned by convention. Since there is no previous committed update, the historical span is empty and the current LoRA update is the only available adaptation direction. This first coefficient should therefore not be interpreted as evidence that the first dataset intrinsically requires full residual commitment.

LibriSpeech-other tends to receive a small residual coefficient when it appears after another speech domain. In Order 2, after TED2, HarMoLoRA selects $\eta = 0.25$. The candidate-loss table confirms that $\eta = 0.25$ gives the lowest merge validation loss, whereas the raw residual candidate $\eta = 1.0$ gives a noticeably higher loss. This indicates that the trained audiobook domain LoRA update contains residual directions that are not necessary for preserving current-task validation performance after merging. In Order 3 and Order 4, LibriSpeech-other receives even smaller coefficients ($\eta = 0.125$), suggesting that, once noisy, presentation, or web speech directions have been committed, near-best read speech validation performance can be preserved with a small additional residual component.

CHiME3 shows a complementary behavior. As a noisy real environment ASR domain, CHiME3 differs more strongly from clean read speech and presentation speech in terms of channel and background noise. Accordingly, it often receives a

larger residual coefficient. In Order 1, where CHiME3 follows LibriSpeech-other, HarMoLoRA selects $\eta = 0.75$. In Order 2, where CHiME3 follows both TED2 and LibriSpeech-other, the method selects $\eta = 0.50$. The candidate loss diagnostic shows that $\eta = 0.75$ achieves the minimum loss for CHiME3, while $\eta = 0.50$ remains within the near-best range and is selected by the conservative rule. Thus, HarMoLoRA does not simply choose the largest noise domain update. It chooses the smallest residual coefficient that preserves near-best current-task merge validation loss after rank-constrained realization.

TED2 receives $\eta = 0.25$ when it appears after previous domains in Orders 1 and 3. TED talks are presentation style speech and are often more structured than crowd-sourced or noisy recordings. The small residual coefficient suggests that, after previous ASR domains have already shaped the model, near-best TED2 validation loss can be preserved with only a limited additional residual correction.

The longer Order 4 stream provides a clearer view of how the historical span evolves. GigaSpeech-YOU receives $\eta = 0.25$ after TED2, suggesting that web video speech requires some residual adaptation but does not require the full raw update to be committed. Common Voice receives $\eta = 0.125$, which is consistent with its heterogeneous speaker and device conditions being partly covered by the already committed TED2, YouTube, and LibriSpeech-other directions. GigaSpeech-POD receives $\eta = 0.0$. This value should be interpreted carefully: $\eta = 0.0$ does not discard the trained podcast update entirely. It selects the zero residual candidate, namely the candidate formed from the historical span projection of the trained podcast update, and this candidate is then realized under the original LoRA rank constraint. This suggests that, at that stage of the stream, the podcast validation loss can be preserved without admitting an additional residual component.

Finally, CHiME3 receives $\eta = 1.0$ at the end of Order 4. This is particularly informative because the preceding stream already contains presentation speech, YouTube speech, read speech, crowd-sourced speech, and podcast speech. The full residual coefficient suggests that the noisy real environment correction learned for CHiME3 lies outside the previously committed span and remains useful after merging. Together with the $\eta = 0.0$ decision for GigaSpeech-POD, this shows that HarMoLoRA is not a monotonic shrinkage method. It

can select a zero residual candidate when an additional residual component is unnecessary for near-best validation loss, and it can fully admit a residual component when the realized merge validation loss supports it.

These dataset- and order-level patterns support the motivation for post-training residual selection in continual ASR. Speech domains are not unrelated tasks: they share phonetic structure, transcription objectives, and decoder side subword space. At the same time, each domain can introduce domain-specific acoustic, channel, and style variations. HarMoLoRA uses merge validation to decide which residual directions are useful for the realized single ASR model, rather than assuming that every raw LoRA update should be fully accumulated.

A.5 Recovering LoRA Factors After Rank-Constrained Realization

This section verifies how an effective rank-constrained update can be stored as LoRA factors under the same convention used throughout the paper:

$$\Delta W = sAB, \quad (42)$$

where $s > 0$, $A \in \mathbb{R}^{d_{\text{out}} \times r}$, and $B \in \mathbb{R}^{r \times d_{\text{in}}}$.

Let the selected effective update for layer ℓ be $M_t^\ell(\eta)$, and let its rank- r truncated SVD be

$$\Pi_r(M_t^\ell(\eta)) = U_r \Sigma_r V_r^\top. \quad (43)$$

We can set

$$\hat{A}_t^\ell = \frac{1}{\sqrt{s_\ell}} U_r \Sigma_r^{1/2}, \quad \hat{B}_t^\ell = \frac{1}{\sqrt{s_\ell}} \Sigma_r^{1/2} V_r^\top. \quad (44)$$

Then

$$s_\ell \hat{A}_t^\ell \hat{B}_t^\ell = U_r \Sigma_r V_r^\top = \Pi_r(M_t^\ell(\eta)). \quad (45)$$

Thus, the factor recovery is exact for the truncated rank- r update. The only approximation is the SVD truncation itself, and that approximation is measured by the realization error reported in Table 7.

A.6 Why Post-Training Residual Selection Fits ASR

ASR domain shifts are structured rather than arbitrary. TED talks, audiobooks, Common Voice recordings, YouTube speech, podcasts, and noisy real environment speech differ in channel, speaking style, background noise, speaker distribution, and lexical distribution. However, they still share the

same transcription objective, overlapping acoustic representations, and the same decoder side sub-word space. This makes continual ASR different from a sequence of unrelated tasks.

This distinction matters for update geometry. In a task sequence with unrelated label spaces, separating task-specific update directions can be a reasonable way to reduce interference. In ASR, however, some update directions are naturally shared. A direction that improves phonetic discrimination, stabilizes decoder behavior, or corrects a common acoustic representation can benefit more than one domain. Forcing all domains into orthogonal low-rank subspaces may therefore waste limited LoRA rank capacity and suppress beneficial transfer.

HarmoLoRA followed a different principle. It first allowed the current domain LoRA branch to learn the new domain, preserving plasticity during adaptation to each task. It then selected how much of the residual update should be admitted to the single persistent model. The candidate loss diagnostics show that near-best current-task merge validation loss could be achieved at residual coefficients smaller than the full raw update. Therefore, the deployed update did not need to absorb every residual direction learned by the current-task LoRA branch.

Overall, HarmoLoRA is not simply an update shrinking method. It is a post-training residual commit selection method that uses the recognition loss of the realized merged ASR model to determine how much residual information should be absorbed. This is aligned with replay-free continual ASR, where the final system uses a single model, does not rely on task identifiers, and must balance domain-specific plasticity with shared speech recognition structure.

A.7 Sensitivity to the Near-Best Tolerance

All additional experiments in Sections A.7–A.12 use LoRA rank 8, whereas the main experiments use rank 4. We evaluate the CHiME3 → LibriSpeech-other → TED2 order and keep the default residual grid $\{0, 0.125, 0.25, 0.5, 0.75, 1.0\}$ fixed while varying the near-best tolerance.

Table 9 reports the sensitivity to the near-best tolerance. HarmoLoRA improves Last WER, Avg WER, BWT, and forgetting over IncLoRA for all four tested tolerance values. In this order, Last WER decreases from 14.73 at $\tau = 0.5$ to 13.57 at $\tau = 0.01$, while Avg WER decreases from 17.52 to 16.90. Forgetting is 0.80 for $\tau \in \{0.5, 0.1, 0.05\}$

Method	τ	Last (↓)	Avg (↓)	BWT (↑)	Forgetting (↓)
IncLoRA	–	16.53	18.66	-1.05	1.75
HarmoLoRA	0.5	14.73	17.52	-0.40	0.80
HarmoLoRA	0.1	14.50	17.47	-0.20	0.80
HarmoLoRA	0.05	14.20	17.13	-0.45	0.80
HarmoLoRA	0.01	13.57	16.90	0.45	0.55

Table 9: Sensitivity to the near-best tolerance at LoRA rank 8.

and decreases to 0.55 at $\tau = 0.01$. The $\tau = 0.01$ setting gives the best result in all four reported metrics for this sensitivity experiment. These results show that the improvement over IncLoRA is preserved across the tested tolerance range and is not specific to the default $\tau = 0.1$ value.

A.8 Joint Sensitivity to the Residual Grid and Tolerance

We next vary the residual coefficient grid and the tolerance jointly with LoRA rank 8 and the same task order. The dense grid adds intermediate coefficients at increments of 0.1, whereas the coarse grid retains only five coefficients.

Table 10 reports the joint sensitivity to the residual coefficient grid and the near-best tolerance. Across the four configurations, Last WER lies between 14.03 and 14.50, a range of 0.47 points, and Avg WER lies between 17.07 and 17.43, a range of 0.36 points. The coarse grid with $\tau = 0.05$ gives the lowest Last and Avg WER, the coarse grid with $\tau = 0.1$ gives the highest BWT, and the dense grid with $\tau = 0.05$ gives the lowest forgetting. Thus, no single configuration is best for every metric. More importantly for the commit rule, all four grid and tolerance combinations remain substantially below the rank 8 IncLoRA baseline in Last and Avg WER. The results therefore indicate that the observed recognition gains do not depend on one particular grid resolution.

A.9 Performance at Each Adaptation Stage for Rank 8

Table 11 reports the evaluation suite after every task. The results after Task 1 are identical because no historical span exists before the first task, so both methods commit the initial CHiME3 update without a residual admission decision. Differences begin after LibriSpeech-other, when previous committed information is available.

After Task 2, HarmoLoRA at $\tau = 0.1$ reduces the three-domain Avg from the IncLoRA value of

Residual coefficient grid	τ	Last ($\downarrow$)	Avg ($\downarrow$)	BWT ($\uparrow$)	Forgetting ($\downarrow$)
$\{0, 0.1, \dots, 0.9, 1.0\}$	0.1	14.17	17.14	-0.25	0.90
$\{0, 0.1, \dots, 0.9, 1.0\}$	0.05	14.50	17.42	0.05	0.65
$\{0, 0.25, 0.5, 0.75, 1.0\}$	0.1	14.43	17.43	0.45	0.90
$\{0, 0.25, 0.5, 0.75, 1.0\}$	0.05	14.03	17.07	-0.30	0.75

Table 10: Joint residual grid and tolerance sensitivity at LoRA rank 8.

Method	Adaptation Stage	CHiME3	LibriSpeech-other	TED2	Avg
IncLoRA	After Task 1 (CHiME3)	34.1	17.5	13.2	21.60
	After Task 2 (LibriSpeech-other)	32.9	9.9	10.7	17.83
	After Task 3 (TED2)	31.5	13.4	4.7	16.53
HarmoLoRA ($\tau = 0.1$)	After Task 1 (CHiME3)	34.1	17.5	13.2	21.60
	After Task 2 (LibriSpeech-other)	29.5	9.1	10.3	16.30
	After Task 3 (TED2)	28.3	10.7	4.5	14.50
HarmoLoRA ($\tau = 0.01$)	After Task 1 (CHiME3)	34.1	17.5	13.2	21.60
	After Task 2 (LibriSpeech-other)	28.2	9.2	9.2	15.53
	After Task 3 (TED2)	26.2	10.3	4.2	13.57

Table 11: WER at each adaptation stage for the rank 8 experiment. Avg is the arithmetic mean of the three displayed domain WERs.

Method	Adaptation Stage	CHiME3	LibriSpeech-other	TED2	Avg
IncLoRA	After Task 1 (CHiME3)	34.1	17.5	13.2	21.60
	After Task 2 (LibriSpeech-other)	32.9	9.9	10.7	17.83
	After Task 3 (TED2)	31.5	13.4	4.7	16.53
	After Task 4 (CHiME3)	29.5	13.3	7.8	16.87
HarmoLoRA ($\tau = 0.1$)	After Task 1 (CHiME3)	34.1	17.5	13.2	21.60
	After Task 2 (LibriSpeech-other)	29.5	9.1	10.3	16.30
	After Task 3 (TED2)	28.3	10.7	4.5	14.50
	After Task 4 (CHiME3)	25.8	9.4	6.7	13.97

Table 12: Cyclic rank 8 evaluation. Avg is the arithmetic mean of the three displayed domain WERs.

17.83 to 16.30, while $\tau = 0.01$ reduces it further to 15.53. At this stage, the two HarmoLoRA settings have nearly identical LibriSpeech-other WER, 9.1 and 9.2, while the stricter setting has lower CHiME3 and TED2 WER. After Task 3, $\tau = 0.01$ has lower WER than $\tau = 0.1$ on all three domains and lowers Avg from 14.50 to 13.57.

The adaptation sequence also shows how the earlier domains change after the final TED2 adaptation. CHiME3 improves from 34.1 after Task 1 to 31.5 with IncLoRA, 28.3 with HarmoLoRA at $\tau = 0.1$, and 26.2 at $\tau = 0.01$. For LibriSpeech-other, WER increases from 9.9 to 13.4 after Task 3 with IncLoRA, a 3.5-point change. The corresponding increases are 1.6 points for HarmoLoRA at $\tau = 0.1$ and 1.1 points at $\tau = 0.01$. These per domain trajectories are consistent with the lower final Avg and forgetting values of the HarmoLoRA models in this order.

A.10 Cyclic Evaluation

We extend the rank 8 stream to CHiME3 $\rightarrow$ LibriSpeech-other $\rightarrow$ TED2 $\rightarrow$ CHiME3. At Task 4, the persistent model is adapted to CHiME3 again using only the current CHiME3 data, while maintaining the same single-model continual learning setting.

Table 12 reports the cyclic evaluation results. After Task 4, HarmoLoRA obtains WERs of 25.8, 9.4, and 6.7 on CHiME3, LibriSpeech-other, and TED2, respectively, yielding an Avg WER of 13.97. The corresponding IncLoRA model state yields WERs of 29.5, 13.3, and 7.8, with an Avg WER of 16.87. Thus, after returning to CHiME3, HarmoLoRA achieves lower WER than IncLoRA on all three evaluation domains.

A.11 Six-Domain Evaluation for Order 5

We additionally evaluate Order 5 with LoRA rank 8 on the six-domain stream CHiME3 $\rightarrow$ GigaSpeech-POD $\rightarrow$ LibriSpeech-other $\rightarrow$ Common Voice $\rightarrow$

Method	Last ($\downarrow$)	Avg ($\downarrow$)	BWT ($\uparrow$)	Forgetting ($\downarrow$)	Selected η
IncLoRA	18.93	18.37	-6.36	6.36	—
HarmoLoRA ($\tau = 0.1$)	14.37	16.44	-1.84	1.84	1.0, 0.5, 0.5, 0.5, 0.75, 0.75

Table 13: Rank 8 results for the six-domain Order 5 stream. The selected HarmoLoRA residual coefficients are listed in task order.

Method	Last ($\downarrow$)	Avg ($\downarrow$)	BWT ($\uparrow$)	Forgetting ($\downarrow$)
LoRA P&M (Qiu et al., 2025)	13.97	17.51	0.20	0.90
C-LoRA (Zhang et al., 2025)	14.60	18.18	-0.85	1.15
GainLoRA (Liang et al., 2025)	14.73	17.74	-0.45	0.85
JANUS-LoRA (Chen et al., 2026)	15.23	18.44	-0.35	0.95
HarmoLoRA ($\tau = 0.1$)	14.50	17.47	-0.20	0.80
HarmoLoRA ($\tau = 0.01$)	13.57	16.90	0.45	0.55

Table 14: Comparison with recent continual learning and continual LoRA methods in the rank 8 ASR setting.

GigaSpeech-YOU $\rightarrow$ TED2. The experiment follows the same replay-free, single-model continual learning setting: at each task, the model accesses only the current-domain data and commits the selected update to one persistent model. HarmoLoRA uses the default tolerance $\tau = 0.1$.

Table 13 reports the results for the six domain Order 5 stream with rank 8. HarmoLoRA reduces Last WER from 18.93 to 14.37, an absolute improvement of 4.56 points, and reduces trajectory level Avg WER from 18.37 to 16.44, an improvement of 1.93 points. BWT improves from -6.36 to -1.84 , while forgetting decreases from 6.36 to 1.84. These correspond to a 4.52-point improvement in BWT and a 4.52-point reduction in forgetting. Although BWT remains negative, its substantially smaller magnitude together with the lower forgetting shows that HarmoLoRA reduces previous task degradation relative to IncLoRA in this six domain stream.

The selected residual coefficients also vary across the stream. Task 1, CHiME3, uses $\eta = 1.0$ by convention because no historical span exists before the first task. HarmoLoRA then selects $\eta = 0.5$ for GigaSpeech-POD, LibriSpeech-other, and Common Voice, followed by $\eta = 0.75$ for GigaSpeech-YOU and TED2. Thus, all tasks after the first use partial residual admission rather than committing the full residual component. Because HarmoLoRA selects the smallest coefficient whose realized merge-validation loss satisfies the nearest criterion, the larger coefficients selected for GigaSpeech-YOU and TED2 indicate that smaller residual admission coefficients did not satisfy the same criterion at these tasks. The resulting coeffi-

cients therefore depend on the current trained update and the historical span available at each stage, rather than following a fixed shrinkage schedule.

A.12 Comparison with Recent Continual LoRA Methods

We additionally reproduce LoRA P&M (Qiu et al., 2025), C-LoRA (Zhang et al., 2025), GainLoRA (Liang et al., 2025), and JANUS-LoRA (Chen et al., 2026) with LoRA rank 8 under the same CHiME3 $\rightarrow$ LibriSpeech-other $\rightarrow$ TED2 ASR order. Since these methods were originally developed in different task or model settings, Table 14 reports our ASR reproductions under a common protocol rather than results copied from the original publications.

With the default $\tau = 0.1$, HarmoLoRA has 0.53-point higher Last WER than LoRA P&M, but 0.04-point lower Avg WER and 0.10-point lower forgetting; LoRA P&M has 0.40-point higher BWT. Against C-LoRA, HarmoLoRA at $\tau = 0.1$ reduces Last and Avg WER by 0.10 and 0.71 points, increases BWT by 0.65 points, and reduces forgetting by 0.35 points. The corresponding improvements over GainLoRA are 0.23 Last WER, 0.27 Avg WER, 0.25 BWT, and 0.05 forgetting, and over JANUS-LoRA they are 0.73 Last WER, 0.97 Avg WER, 0.15 BWT, and 0.15 forgetting. The $\tau = 0.01$ sensitivity setting obtains the best value in all four columns. Because $\tau = 0.01$ was evaluated as a sensitivity setting, the fixed $\tau = 0.1$ configuration remains the primary reference for the main HarmoLoRA design.