A Study on Regularization-Based Continual Learning Methods for Indic ASR

Anonymous ACL submission

Abstract

India's linguistic diversity challenges inclusive Automatic Speech Recognition (ASR) system development. Traditional multilingual models, requiring simultaneous access to all language data, are impractical due to sequential data arrival and privacy constraints. Continual Learning (CL) enables models to learn new languages sequentially without catastrophically forgetting prior knowledge. This paper investigates CL for ASR on Indian languages using the subset of the `IndicSUPERB` benchmark. We employ a Conformer-based hybrid RNNT-CTC model, initially pretrained on Hindi, which is subsequently trained incrementally on eight additional Indian languages, for a sequence of nine languages in total. We evaluate three prominent regularization and distillation-based CL strategies: Elastic Weight Consolidation (EWC), Memory Aware Synapses (MAS), and Learning without Forgetting (LwF), chosen for their suitability in no-replay, privacy-conscious scenarios. Performance is analyzed using Word Error Rate (WER) for both RNNT and CTC paths on clean/noisy data, and knowledge retention via Backward Transfer. We explore varying training epochs (1, 2, 5 and 10) per task. Results, compared against naive fine-tuning, demonstrate CL's efficacy in mitigating forgetting for scalable ASR in diverse Indian languages under realistic constraints. The code is available at https://anonymous.4open.science/r/Indic-CL-ASR-9FF7

1 Introduction

India's extensive linguistic diversity poses significant hurdles for developing comprehensive Automatic Speech Recognition (ASR) systems (Zhong et al., 2024). Traditional multilingual models, typically trained on aggregated datasets (Bai et al., 2021), are ill-suited for real-world scenarios characterized by incremental data availability for low-resource languages, high computational costs of retraining, and data privacy concerns (Della Libera et al., 2024). Continual Learning (CL), or lifelong learning (Ring, 1997; De Lange et al., 2021), offers a paradigm to address these issues by enabling models to learn new tasks (languages) sequentially while preserving previously acquired knowledge. The primary challenge in CL is catastrophic forgetting: the tendency of models to lose performance on past tasks when trained on new ones (McCloskey and Cohen, 1989). Mitigating this is crucial for successful CL application (Kirkpatrick et al., 2017; Goodfellow et al., 2015). This work applies CL to multilingual ASR for Indian languages using the subset of the `IndicSUPERB` benchmark (Jain et al., 2024). We start with the `indicconformer` model (a Conformer-based (Gulati et al., 2020) hybrid RNNT-CTC (Burchi et al., 2024; Graves, 2012; Graves et al., 2006) system pretrained on Hindi using NeMo (Harper et al.)) and incrementally train it on nine additional Indian languages: Bengali, Marathi, Telugu, Tamil, Urdu, Gujarati, Kannada, and Odia. We investigate three established CL strategies: Elastic Weight Consolidation (EWC) (Aich, 2021), Memory Aware Synapses (MAS) (Aljundi et al., 2018), and Learning without Forgetting (LwF) (Li and Hoiem, 2017). These regularization and distillation methods are chosen because architecture-based approaches can inflate model size, and memory-based methods often violate realistic no-replay and privacy constraints (Rebuffi et al., 2017; Lopez-Paz and Ranzato, 2022). Our experiments evaluate WER on clean and noisy data for both RNNT and CTC paths, and Backward Transfer to quantify forgetting, also varying training epochs per language. In summary, our contributions include: (1) the first comprehensive study of CL for ASR across diverse Indian languages (2) systematic evaluation of EWC, MAS, and LwF under realistic constraints, and (3) detailed analysis of WER and knowledge retention across training regimes to guide practical deployment.

2 Related Work

Continual Learning (CL) aims to enable AI systems to learn incrementally from a sequence of tasks without catastrophically forgetting prior knowledge. Key approaches include (Wang et al., 2024) regularization-based methods (e.g., EWC, which penalizes changes to parameters important for past tasks based on the Fisher Information Matrix; MAS, which uses the gradient of the squared L2 norm (Hoerl and Kennard, 1970) of the model's output; SI (Zenke et al., 2017)), rehearsal-based methods (replaying past data) (Chaudhry et al., 2019), and architecture-based methods (dynamically modifying model structure). Applying CL to Automatic Speech Recognition (ASR) is challenging due to sequence variability, acoustic diversity, and linguistic complexity, especially when sequentially learning new languages in low-resource settings, common for many Indian languages. Hybrid CTC-RNNT models (Hori et al., 2017), prevalent in modern ASR, offer multiple avenues for CL integration. Our work explores EWC, MAS, and (LwF), which employs knowledge distillation to preserve the previous model's outputs on new data without storing old data. We utilize the subset of the `IndicSUPERB` benchmark (Jain et al., 2024), which provides standardized speech datasets for multiple Indian languages (including clean/noisy splits), and the `indicconformer`, a state-of-the-art Conformer-based hybrid RNNT-CTCmodel pre-trained on Hindi, as our base model and evaluation framework.

3 Benchmark Design

Our benchmark simulates realistic constraints for continual learning in multilingual ASR using the subset IndicSUPERB dataset. Each Indian language is treated as a separate task, forming a sequence of nine tasks beginning with Hindi (T_1), followed by Bengali, Marathi, Telugu, Tamil, Urdu, Gujarati, Kannada, and Odia (T_2 to T_9). All tasks are presented in a low-resource setting, with only 3000 training utterances per language (2000 clean and 1000 noisy). The model is trained sequentially using only the current task's data D_k, enforcing a strict no-data-replay constraint. Training, validation, and test sets contain both clean and noisy samples, with test sets comprising 200 clean and 200 noisy utterances per language. Word Error Rate (WER) is evaluated separately on clean and noisy test splits using both RNNT and CTC decoding paths. To explore the trade-off between adaptation speed, accuracy on new tasks, and knowledge retention, we experiment with 1, 2, 5, and 10 training epochs per task. We benchmark performance against a naive sequential fine-tuning baseline. Further details on task formulation, model architecture, dataset construction and experimentation setup are provided in Appendix A.1, Appendix A.3, Appendix A.2 and Appendix A.5.

4 Evaluation Metrics

- **Word Error Rate (WER):** A commonly used metric in automatic speech recognition (Goldwater et al., 2010) and is expressed as a decimal fraction ranging from 0 to 1. WER is evaluated on all previously learned tasks after each new task is completed. Lower WER indicates better performance.

- **Average Performance:** After training on task T_k, the average WER across all tasks T_1 to T_k is given by:

$$\text{AvgWER}_k = \frac{1}{k} \sum_{i=1}^{k} W_{k,i}$$

where $W_{k,i}$ denotes the WER on task T_i after learning task T_k. Lower AvgWER indicates better overall retention and adaptation.

- **Backward Transfer (BWT):** Quantifies the influence of learning new tasks on the performance of previously learned ones. After task T_k, BWT is defined as:

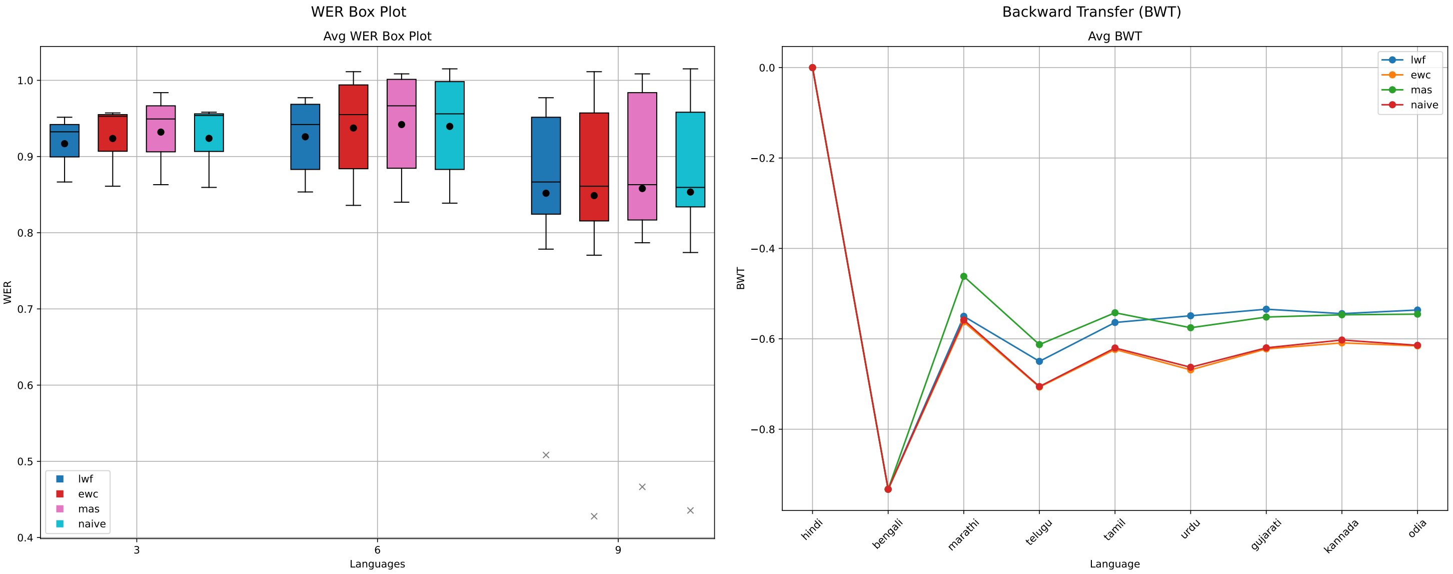
$$\text{BWT}_k = \frac{1}{k-1} \sum_{i=1}^{k-1} (\text{Acc}_{k,i} - \text{Acc}_{i,i})$$

where $\text{Acc}_{k,i} = 1 - W_{k,i}$ is the accuracy on task T_i after learning task T_k, and $\text{Acc}_{i,i} = 1 - W_{i,i}$ is the accuracy on task T_i immediately after it was learned. Higher BWT indicates better retention and less forgetting.

5 Experiments and Results

5.1 Observations

CTC Benchmarking As shown in Figure 1, the average WER across tasks reveals a clear ranking among methods. LwF achieves the best overall performance, followed by EWC, then MAS, with



(a) CTC Box (b) CTC BWT

Figure 1: CTC Benchmark – Box and BWT Plots.

naive fine-tuning performing the worst. This ranking is particularly evident in short and medium task horizons. For longer sequences, however, the performance gap between methods narrows considerably. Naive fine-tuning, in particular, produces the highest WER maxima across tasks. When analyzing backward transfer (BWT), MAS performs best in short sequences, while LwF excels in medium-length tasks. For longer sequences, both MAS and LwF converge to similar average BWT values, whereas EWC and naive fine-tuning fall behind.

RNN-T Benchmarking Figure 9 shows that RNN-T (Xu et al., 2024) consistently outperforms CTC in WER across all continual learning strategies. Among these, EWC achieves the lowest WER across task lengths, demonstrating strong performance retention on the current task. However, this benefit comes at a cost: EWC exhibits the worst BWT of all methods, even lower than that of naive fine-tuning, indicating substantial forgetting. MAS shows some improvement in BWT for medium-length sequences, but for longer horizons, BWT scores deteriorate across all methods except EWC, eventually becoming nearly indistinguishable.

General Comparison of CL Methods under Noisy Settings In noisy conditions (Figure 2), both LwF and MAS outperform EWC and the naive baseline in BWT, suggesting better retention of prior knowledge. Interestingly, noise appears to improve backward transfer, likely due to regularization effects. However, this improvement comes with a trade-off: WER increases, and models perform better on clean audio in absolute terms. This contrast indicates that noise can enhance stability, by reducing forgetting, while simultaneously impairing plasticity, by diminishing learning precision, which is reflected in the higher WER.

WER Performance Analysis Figures 3 and 4 present WER trends over increasing task lengths. Evaluations are averaged over the last two and current tasks, categorized as short (1–3), medium (1–6), and long (1–9). In general, models perform better with clean data. Among the methods, LwF consistently maintains WER below 1.0, with high stability indicated by narrow shaded variance regions.

Interestingly, the upper bounds of noisy WER for LwF are comparable to the maxima seen under clean conditions. This can be attributed to its distillation-based loss, which prevents overfitting to noisy inputs by anchoring the model to previous predictions. MAS follows a similar pattern, though with slightly lower stability. EWC occasionally achieves better minimum WERs, particularly for short tasks, but continues to show poor BWT. The naive method performs surprisingly well in short sequences but fails to retain knowledge over longer horizons. Overall, LwF demonstrates the effectiveness of knowledge distillation in maintaining a balance between acquiring new knowledge and retaining previous learning. For longer sequences, average WER tends to decline, possibly due to simpler language characteristics in later tasks.

EWC Ablation Studies In Figure 5, we examine the impact of different regularization strengths in EWC by testing $\lambda_{\text{EWC}} \in 5, 10$. While both values yield similar outcomes, $\lambda_{\text{EWC}} = 10$ leads to slightly better WER in medium and long tasks, though the benefit is minimal in short tasks. BWT trends (Figure 8) for both values remain close to those of the naive baseline, suggesting limited ability to retain performance on earlier tasks. Additionally, results from epoch-wise ablation (Figure 11) show that increasing training epochs reduces WER, with the best results achieved at epoch 10. However, BWT steadily declines with more epochs (Figure 14), confirming the stability-plasticity trade-off: improved learning on new tasks often leads to increased forgetting of previous ones.

LwF Ablation Studies As shown in Figure 6, adjusting the distillation weight (α_{KD}) significantly impacts LwF's performance. A higher value of 0.5 severely limits the model's ability to learn new tasks, resulting in WERs close to 1.0 across all horizons thus worse than naive fine-tuning for short sequences. In contrast, $\alpha_{\text{KD}} = 0.1$ strikes a better balance, achieving WER comparable to or better than naive fine-tuning while maintaining much stronger BWT. As shown in Figure 8, the 0.5 configuration yields the highest BWT, primarily because the model barely updates and effectively freezes previous knowledge. The 0.1 setting enables more meaningful learning while controlling forgetting.

Epoch-wise trends (Figures 10 and 14) are consistent with those observed in EWC. Increasing the epochs improves WER but worsens BWT.

MAS Ablation Studies In Figure 7, we compare MAS with regularization weights α_{ctx} of 0.3 and 1.0. The stronger setting of 1.0 consistently achieves better WER and shows more stable variance across tasks. Its shaded performance region closely overlaps with that of naive fine-tuning, though with lower dispersion. When examining BWT (Figure 8), the 0.3 configuration performs better, matching LwF in retaining knowledge.

As with the other methods, MAS exhibits the stability-plasticity trade-off: increasing epochs (Figure 12) lowers WER but leads to worsening BWT (Figure 14). This consistent trend across methods emphasizes the fundamental challenge in continual learning of effectively balancing the acquisition of new information with the retention of existing knowledge.

6 Discussion

Our findings show that LwF and MAS generally offer better BWT in noisy ASR, indicating superior retention of prior languages. The inverse link between noise-driven BWT improvement and WER degradation suggests noise acts as an implicit regularizer, improving retention at the cost of transcription accuracy. LwF's consistently low and stable WER, especially in longer task sequences, highlights its distillation-based regularization effectiveness in noisy settings by preventing over-adaptation. In contrast, EWC, while competitive in shorter tasks or with RNN-T, often showed poor BWT, particularly with RNN-T, indicating weight consolidation is less effective for complex recurrent models or sequential multilingual learning.

Ablation studies confirmed the stability-plasticity dilemma. Longer training improves current task WER but worsens BWT. Stronger regularization improves BWT but hinders new learning, while weaker regularization enhances plasticity but increases forgetting. Comparing CTC and RNN-T, RNN-T achieved better WER but worsened catastrophic forgetting, especially for EWC. The decline of BWT in long RNN-T sequences, except for EWC, highlights challenges for current CL methods with advanced ASR models over extended tasks. Notably, despite CL, absolute WER during new task learning remains suboptimal for practical use, underscoring the difficulty in balancing plasticity and retention and the early stage of CL in ASR.

7 Conclusion

This study shows that while LwF and MAS can improve BWT in noisy, multi-language ASR compared to baselines and EWC, a fundamental tradeoff persists. Noise appears to aid BWT, possibly as a regularizer, but consistently degrades WER. LwF offered the most balanced performance with stable, low WER and good BWT for longer sequences. The stability-plasticity dilemma was pervasive: efforts to improve new task learning typically increased forgetting. RNN-T models, while delivering superior WER, amplified catastrophic forgetting. Importantly, even with CL, overall WER during new language learning often remains too high for practical deployment. This signals that current CL methods are not yet complete solutions and that CL in ASR requires further investigation for real-world viability.

8 Limitations

While our work offers valuable insights into continual learning (CL) for multilingual ASR under noise, several limitations must be acknowledged. First, the study does not systematically investigate the impact of language ordering on performance. Since language sequence can significantly influence both task difficulty and forgetting dynamics, this is a key variable requiring further exploration. Second, our findings are constrained to the specific datasets, noise profiles, and ASR architectures (CTC and RNN-T) evaluated. As such, the extent to which these results generalize to other languages, domains, or ASR models (e.g., Transformer-based architectures) remains uncertain.

9 Future Work

To advance CL for ASR towards practical applications, future work should explore:

- **Federated learning frameworks** (Bharati et al., 2022) to address privacy and simulate realistic distributed ASR deployment.
- Transitioning to **online learning paradigms** where data arrives as a continuous stream, reflecting many real-world ASR use-cases and posing new challenges for CL algorithm efficiency (Harun et al., 2023) and adaptability.
- The resilience and adaptation of CL strategies in **adversarial settings** (Ebrahimi et al., 2020) to develop more secure and reliable systems.
- Developing **novel CL techniques** specifically tailored to speech's sequential nature and modern ASR model intricacies (e.g., RNN-T) to better overcome the stability-plasticity dilemma and achieve deployment-ready performance.

References

Abhishek Aich. 2021. Elastic weight consolidation (ewc): Nuts and bolts. *arXiv preprint arXiv:2105.04093*.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154.

Junwen Bai, Bo Li, Yu Zhang, Ankur Bapna, Nikhil Siddhartha, Khe Chai Sim, and Tara N. Sainath. 2021. Joint unsupervised and supervised training for multilingual asr. *Preprint*, arXiv:2111.08137.

Subrato Bharati, M. Rubaiyat Hossain Mondal, Prajoy Podder, and V.B. Surya Prasath. 2022. Federated learning: Applications, challenges and future directions. *International Journal of Hybrid Intelligent Systems*, 18(1–2):19–35.

Maxime Burchi, Krishna C Puvvada, Jagadeesh Balam, Boris Ginsburg, and Radu Timofte. 2024. Multilingual audio-visual speech recognition with hybrid ctc/rnn-t fast conformer. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10211–10215. IEEE.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with a-gem. *Preprint*, arXiv:1812.00420.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385.

Luca Della Libera, Pooneh Mousavi, Salah Zaiem, Cem Subakan, and Mirco Ravanelli. 2024. Cl-masr: A continual learning benchmark for multilingual asr. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. 2020. Adversarial continual learning. *Preprint*, arXiv:2003.09553.

Sharon Goldwater, Dan Jurafsky, and Christopher D. Manning. 2010. Which words are hard to recognize? prosodic, lexical, and disfluency factors that increase speech recognition error rates. *Speech Communication*, 52(3):181–200.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2015. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *Preprint*, arXiv:1312.6211.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. *Preprint*, arXiv:2005.08100.

Eric Harper, Somshubra Majumdar, Oleksii Kuchaiev, Li Jason, Yang Zhang, Evelina Bakhturina, Vahid Noroozi, Sandeep Subramanian, Koluguri Nithin, Huang Jocelyn, Fei Jia, Jagadeesh Balam, Xuesong Yang, Micha Livne, Yi Dong, Sean Naren, and Boris Ginsburg. NeMo: a toolkit for Conversational AI and Large Language Models.

Md Yousuf Harun, Jhair Gallardo, Tyler L. Hayes, and Christopher Kanan. 2023. How efficient are today's continual learning algorithms? *Preprint*, arXiv:2303.18171.

Arthur E. Hoerl and Robert W. Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan. 2017. Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm. *Preprint*, arXiv:1706.02737.

Sparsh Jain, Ashwin Sankar, Devilal Choudhary, Dhairya Suman, Nikhil Narasimhan, Mohammed Safi Ur Rahman Khan, Anoop Kunchukuttan, Mitesh M Khapra, and Raj Dabre. 2024. Bhasaanuvaad: A speech translation dataset for 13 indian languages. *arXiv preprint arXiv:2411.04699*.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *Preprint*, arXiv:1606.09282.

David Lopez-Paz and Marc'Aurelio Ranzato. 2022. Gradient episodic memory for continual learning. *Preprint*, arXiv:1706.08840.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. icarl: Incremental classifier and representation learning. *Preprint*, arXiv:1611.07725.

Mark B Ring. 1997. Child: A first step towards continual learning. *Machine Learning*, 28(1):77–104.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Hainan Xu, Fei Jia, Somshubra Majumdar, Shinji Watanabe, and Boris Ginsburg. 2024. Multi-blank transducers for speech recognition. *Preprint*, arXiv:2211.03541.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR.

Tianyang Zhong, Zhenyuan Yang, Zhengliang Liu, Ruidong Zhang, Yiheng Liu, Haiyang Sun, Yi Pan, Yiwei Li, Yifan Zhou, Hanqi Jiang, and 1 others. 2024. Opportunities and challenges of large language models for low-resource languages in humanities research. *arXiv preprint arXiv:2412.04497*.

A Appendix

A.1 Problem Formulation

We formulate the continual learning (CL) problem in multilingual ASR as a sequential learning setup. Let $D = \{D_1, D_2, ..., D_N\}$ denote a sequence of datasets, each corresponding to a task T_k (i.e., language k). Each dataset $D_k = \{(x_{kj}, y_{kj})\}$ contains speech utterances x_{kj} and transcriptions y_{kj}. The goal is to train an ASR model $M(\theta)$ over tasks $T_1, ..., T_N$ such that it learns the current task well while preserving performance on previous tasks.

During training on task T_k, only data D_k is accessible. A naive fine-tuning approach minimizes the loss for task T_k starting from the parameters θ_{k-1} obtained from the previous task:

$$\theta_k = \arg\min_{\theta} L_k(\theta),$$

where $L_k(\theta)$ is the task-specific loss composed of a weighted sum of RNNT and CTC objectives. However, such fine-tuning often causes *catastrophic forgetting*, where performance degrades significantly on previously learned tasks.

To address this, we integrate three regularization-based CL methods into our training pipeline:

- **Elastic Weight Consolidation (EWC):** Prevents drift on important parameters by adding a quadratic penalty based on a Fisher Information matrix estimated after each task. The updated loss becomes:

$$L_{\text{total}} = L_k(\theta) + \lambda_{\text{EWC}} \sum_j F_j(\theta_j - \theta_j^*)^2,$$

where F_j is the accumulated Fisher importance and θ^* are parameters from the previous task.

- **Memory Aware Synapses (MAS):** Estimates importance via gradients of the squared norm of model outputs (logits) and adds a similar penalty:

  $$L_{\text{total}} = L_k(\theta) + \lambda_{\text{MAS}} \sum_j \Omega_j(\theta_j - \theta_j^*)^2,$$

  where Ω_j is the importance computed from absolute gradients w.r.t. combined RNNT and CTC output activations.

- **Learning without Forgetting (LwF):** Adds a distillation loss to encourage the current model to produce similar outputs as the frozen model from the previous task:

  $$L_{\text{total}} = (1 - \alpha) \cdot L_k(\theta) + \alpha \cdot L_{\text{distill}},$$

  where L_{distill} is a weighted combination of KL divergence or MSE between the current and previous model's RNNT and CTC outputs on current task data.

In our setup:

- Tasks $T_1 \ldots T_9$ correspond to the 9 Indian languages in `IndicSUPERB`.
- Only D_k is available while training on task T_k.
- The model $M(\theta_0)$ is initialized from a Hindi-pretrained `indicconformer`.
- The base loss L_k is:

  $$L_k(\theta) = (1 - w_{\text{CTC}}) \cdot L_{\text{RNNT}} + w_{\text{CTC}} \cdot L_{\text{CTC}}.$$

This formulation allows us to balance plasticity (learning new tasks) and stability (retaining performance on past tasks) through principled integration of CL techniques.

A.2 Dataset

We conduct our experiments using the `IndicSUPERB` benchmark, which originally encompasses 11 Indian languages. For this study, we focus on nine languages: Hindi (hi), Bengali (bn), Marathi (mr), Telugu (te), Tamil (ta), Urdu (ur), Gujarati (gu), Kannada (kn), and Odia (or). These languages cover both the Indo-Aryan and Dravidian families, ensuring linguistic diversity.

To simulate a low-resource scenario, we utilize a subset of 3,000 training utterances per language, composed of 2,000 clean and 1,000 noisy samples. The validation and test sets each consist of 400 utterances, evenly split between clean and noisy conditions. This consistent setup allows us to rigorously evaluate model performance under constrained data conditions across multiple languages.

A.3 Model Architecture

Our automatic speech recognition system (`indicconformer`) is built around a hybrid architecture that combines a Conformer-based encoder with both Recurrent Neural Network Transducer (RNNT) and Connectionist Temporal Classification (CTC) objectives using NeMo (Harper et al.). The Conformer encoder effectively captures speech features by integrating convolutional layers to model local dependencies alongside self-attention mechanisms for global context.

The RNNT component models output sequences in an end-to-end fashion, composed of an encoder, a prediction network that autoregressively generates hypotheses based on previous tokens, and a joint network that fuses these signals. This structure inherently manages acoustic modeling and alignment without requiring explicit segmentation.

In parallel, the CTC loss facilitates training without frame-level alignment by introducing a blank token and summing probabilities over all valid alignments. Often used as an auxiliary objective, CTC guides the encoder towards robust and stable feature representations.

We train the model by jointly optimizing the RNNT and CTC losses, combining them in a weighted sum:

$$L_{\text{base}} = (1 - w_{\text{CTC}}) \cdot L_{\text{RNNT}} + w_{\text{CTC}} \cdot L_{\text{CTC}}$$

where w_{CTC} is the weight for the CTC loss.

A.4 Continual Learning Methods Implementation

To mitigate forgetting in continual learning, we augment the base loss with regularization losses depending on the method used.

A.4.1 Learning without Forgetting (LwF)

LwF employs a knowledge distillation loss using KL-divergence (Kullback and Leibler, 1951) that encourages the current model to mimic the outputs of the frozen previous model on the new data. Distillation is applied separately on the RNNT logits and CTC output probabilities.

$$L_{\text{dist}}^{\text{RNNT}} = \text{DistillationLoss}\big(O_{\text{RNNT}}(\theta),\, O_{\text{RNNT}}(\theta^*)\big),$$
$$L_{\text{dist}}^{\text{CTC}} = \text{DistillationLoss}\big(O_{\text{CTC}}(\theta),\, O_{\text{CTC}}(\theta^*)\big),$$

where O_{RNNT} and O_{CTC} denote the outputs (logits or probabilities) of the current and frozen models respectively.

The total distillation loss is a weighted sum:

$$L_{\text{dist}} = (1-\alpha_{\text{ctx}}) \cdot L_{\text{dist}}^{\text{RNNT}} + \alpha_{\text{ctx}} \cdot L_{\text{dist}}^{\text{CTC}},$$

with $\alpha_{\text{ctx}} \in [0,1]$ balancing between RNNT and CTC distillation.

Finally, the full training loss is:

$$L_{\text{total}} = (1-\alpha_{\text{KD}}) \cdot L_{\text{base}} + \alpha_{\text{KD}} \cdot L_{\text{dist}},$$

where $\alpha_{\text{KD}} \in [0,1]$ controls the strength of the knowledge distillation regularization.

A.4.2 Memory Aware Synapses (MAS)

MAS estimates parameter importance by measuring the sensitivity of the squared norm of the model's outputs to each parameter. This is done separately for the CTC decoder and the RNNT joint network logits.

First, compute the squared logit norms and average over the batch:

$$L_{\text{CTC\_logits}} = \frac{1}{B}\sum_{b=1}^{B} \left\|\mathbf{z}_{\text{CTC}}^{(b)}\right\|_2^2$$

where $\mathbf{z}_{\text{CTC}}^{(b)}$ are the flattened CTC decoder logits for batch element b.

Similarly, compute the average squared norm over the stored RNNT joint network logits:

$$L_{\text{RNNT\_logits}} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{B}\sum_{b=1}^{B} \left\|\mathbf{z}_{\text{RNNT},n}^{(b)}\right\|_2^2,$$

where $\mathbf{z}_{\text{RNNT},n}^{(b)}$ is the flattened joint logits tensor stored at step n, and N is the total number of stored logits.

Combine these with a weighting factor $\alpha_{\text{ctx}} \in [0,1]$:

$$L_{\text{logits}} = (1-\alpha_{\text{ctx}}) \cdot L_{\text{RNNT\_logits}} + \alpha_{\text{ctx}} \cdot L_{\text{CTC\_logits}}.$$

Perform backpropagation on L_{logits} to obtain gradients $\nabla_{\theta_j} L_{\text{logits}}$. Then, update parameter importance values as the accumulated absolute gradients:

$$\Omega_j \leftarrow \Omega_j + \left|\frac{\partial L_{\text{logits}}}{\partial \theta_j}\right|.$$

Finally, the MAS regularization penalty is computed as:

$$L_{\text{MAS}} = \lambda_{\text{MAS}} \sum_j \Omega_j (\theta_j - \theta_j^*)^2,$$

where λ is the MAS regularization strength, and θ_j^* are the parameters saved after the previous task.

The full training loss is:

$$L_{\text{total}} = L_{\text{base}} + L_{\text{MAS}}.$$

A.4.3 Elastic Weight Consolidation (EWC)

EWC mitigates catastrophic forgetting by penalizing changes to parameters deemed important for previously learned tasks. Importance is quantified using the diagonal of the Fisher Information Matrix.

After task T_i, the diagonal Fisher is estimated as:

$$F_{i,j} = \mathbb{E}_{x\sim D_i}\left[\left(\frac{\partial L_i(\theta)}{\partial \theta_j}\right)^2\right],$$

where $F_{i,j}$ denotes the importance of parameter θ_j and is computed by averaging squared gradients over the dataset D_i.

To accumulate importance across tasks, we update the consolidated Fisher with a decay factor γ:

$$F_{\text{consol},i} = \gamma \cdot F_{\text{consol},i-1} + F_i.$$

This allows older tasks' importance to gradually decay while emphasizing more recent tasks.

During training on a new task, the EWC penalty is added to the base loss:

$$L_{\text{EWC}} = \lambda_{\text{EWC}} \sum_j F_{\text{consol},j}(\theta_j - \theta_j^*)^2,$$

where θ_j^* are the parameter values saved after the previous task, and λ controls the regularization strength.

The full training loss becomes:

$$L_{\text{total}} = L_{\text{base}} + L_{\text{EWC}}.$$

In practice, the penalty gradient with respect to each parameter θ_j is computed as:

$$\frac{\partial L_{\text{EWC}}}{\partial \theta_j} = 2\lambda \cdot F_{\text{consol},j}(\theta_j - \theta_j^*),$$

which directly enters the optimization step during gradient update.

A.4.4 Summary of Hyperparameters

- w_{CTC}: Weight of CTC loss in the base loss.
- α_{KD}: Weight of the knowledge distillation loss in LwF.
- α_{ctx}: Balancing weight between RNNT and CTC components in distillation and MAS.
- λ: Regularization strength for MAS and EWC.

A.5 Experimental Setup

All experiments are conducted on an NVIDIA V100 GPU using the XXX supercomputer SLURM cluster. Each run took about 13 hours to 3 days depending on the ablation hyper parameters. We initialize our models with the `indicconformer` pretrained on Hindi (`ai4bharat/indicconformer_stt_hi_hybrid_rnnt_large`) using NeMo (Harper et al.), providing a strong starting point for multilingual speech recognition. The model used in our experiments consists of approximately 130 million parameters. The dataset consists of the `IndicSUPERB` benchmark split across nine Indian languages.

Our continual learning experiments follow a fixed sequence of tasks: Hindi → Bengali → Marathi → Telugu → Tamil → Urdu → Gujarati → Kannada → Odia. For each new task, the model is initialized from the previously trained model and trained exclusively on the current language's data (3,000 samples: 2,000 clean and 1,000 noisy).

Training is performed for varying numbers of epochs (1, 2, 5, and 10) to evaluate how training duration impacts model performance and forgetting. Optimization is done using Adam (Kingma, 2014) with a learning rate of 1×10^{-4}.

We apply the following continual learning parameters:

- **Elastic Weight Consolidation (EWC)** with $\lambda_{\text{MAS}} \in \{10, 5\}$ and $\gamma = 1.0$
- **Memory Aware Synapses (MAS)** with $\lambda_{\text{MAS}} = 1$ and $\alpha_{\text{ctx}} \in \{0.3, 1.0\}$
- **Learning without Forgetting (LwF)** with $\alpha_{\text{KD}} \in \{0.1, 0.5\}$ and $\alpha_{\text{ctx}} = 0.3$

The base model is trained using a weighted combination of RNNT and CTC losses with weights:

$$w_{\text{RNNT}} = 0.7, \quad w_{\text{CTC}} = 0.3$$

As a baseline, naive fine-tuning (training on each new task without any continual learning strategy) is also evaluated.

After training on each task T_k, we evaluate the model on the test sets of all tasks from T_1 through T_k. This allows us to compute Word Error Rate (WER) and continual learning metrics such as average accuracy, forgetting, and retention. Hyperparameters and optimization settings are kept consistent across all methods and tasks to ensure fair and reproducible comparisons.

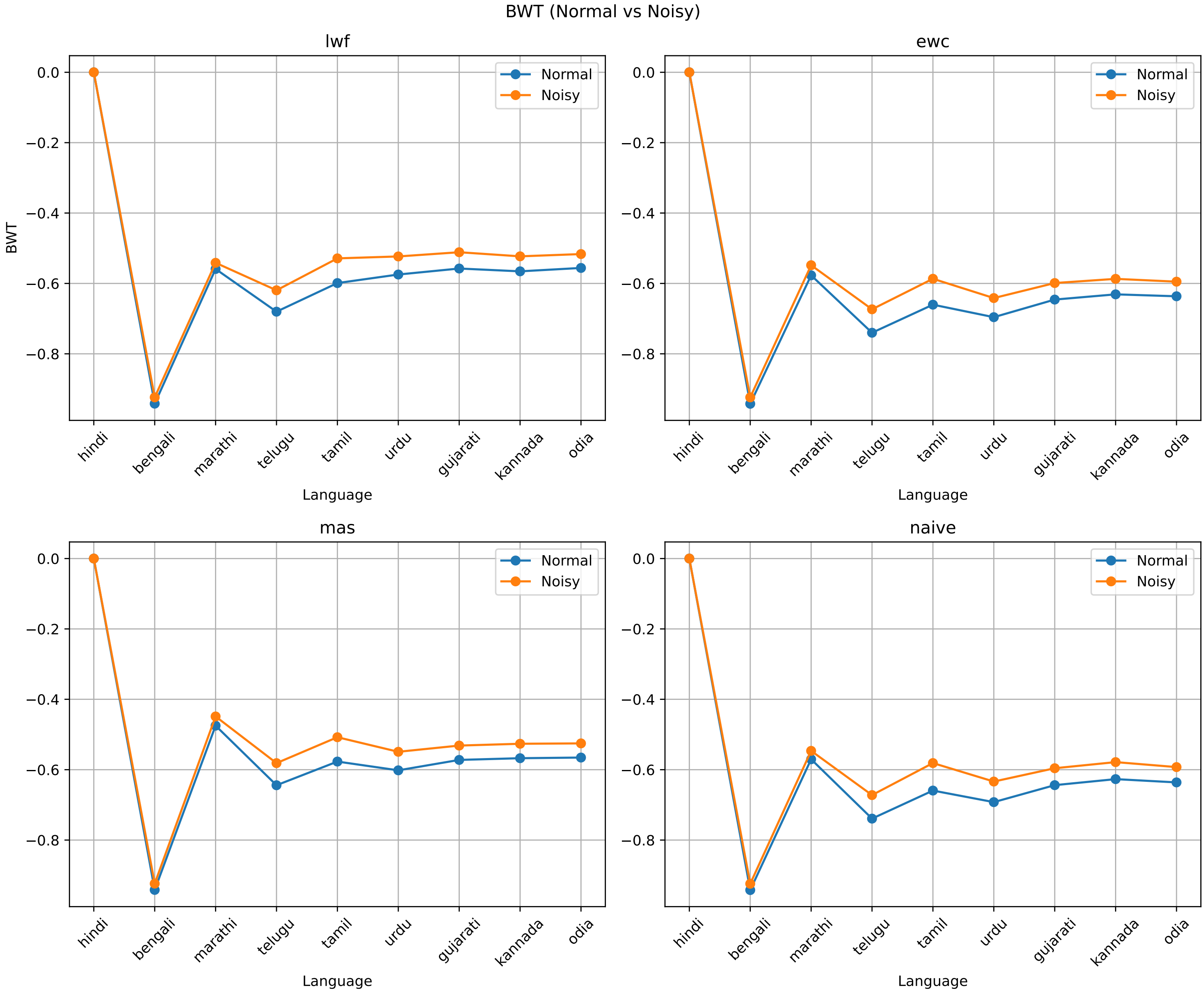


Figure 2: All comparison noisy BWT plot

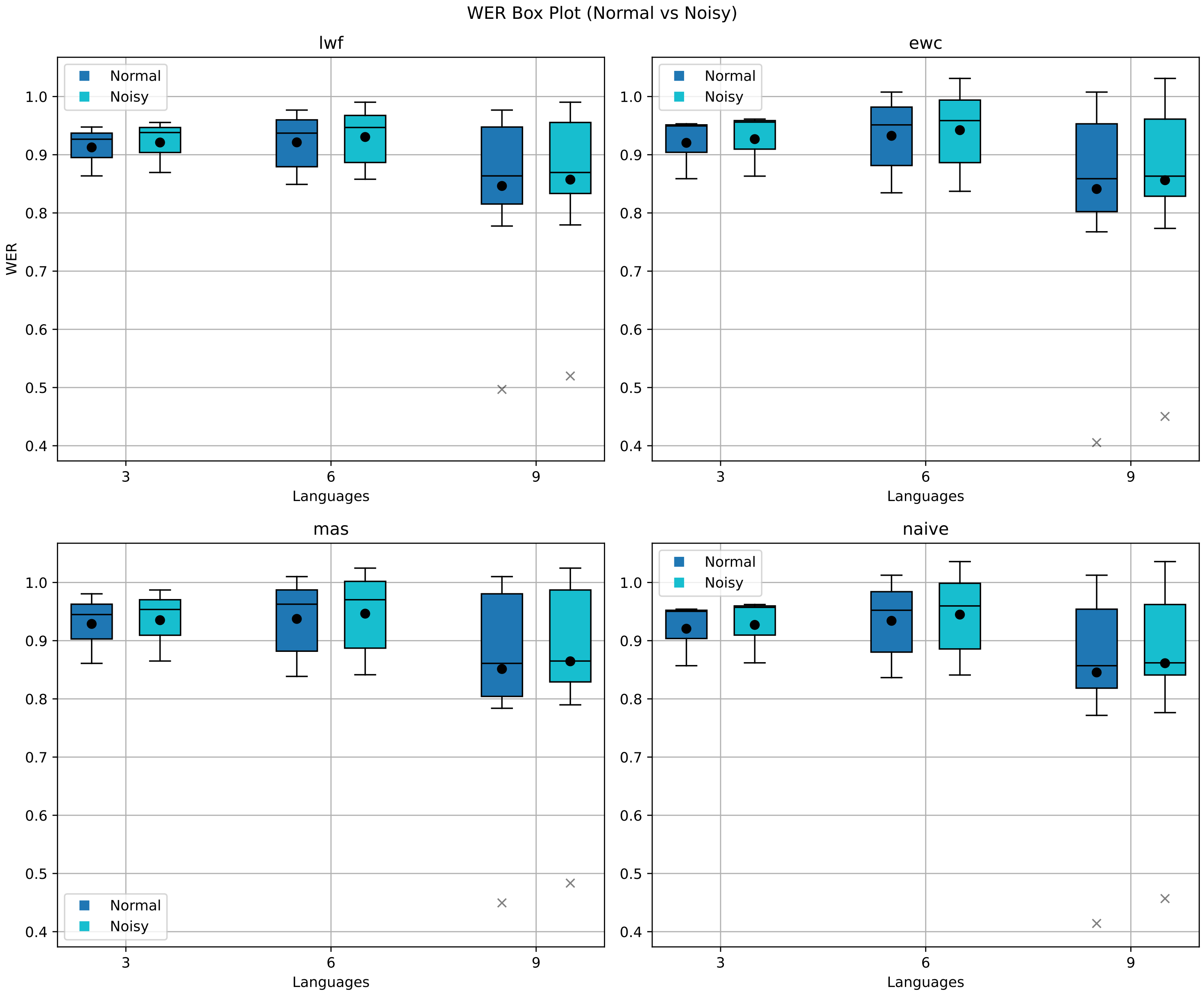


Figure 3: All comparison noisy WER box plot

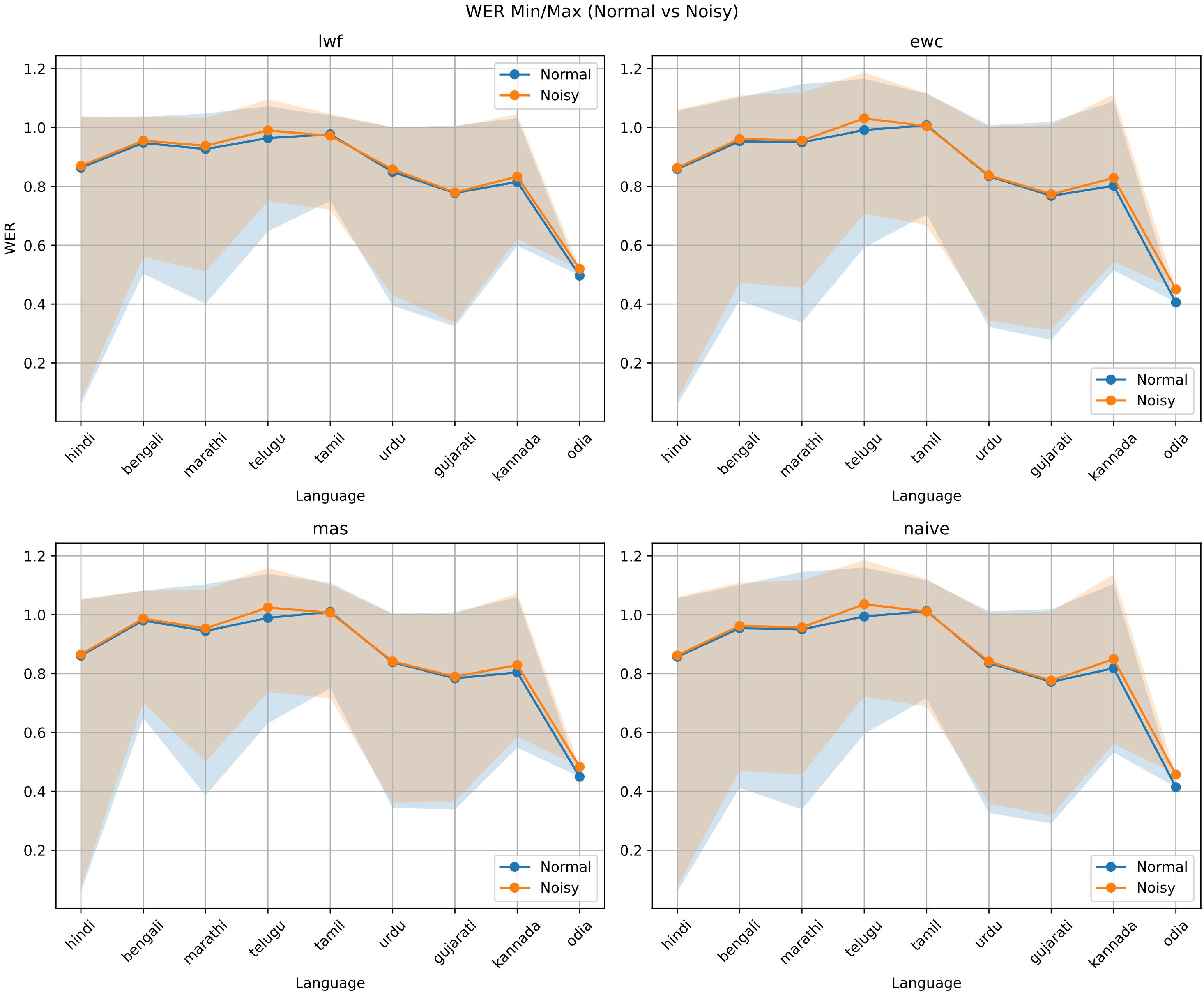
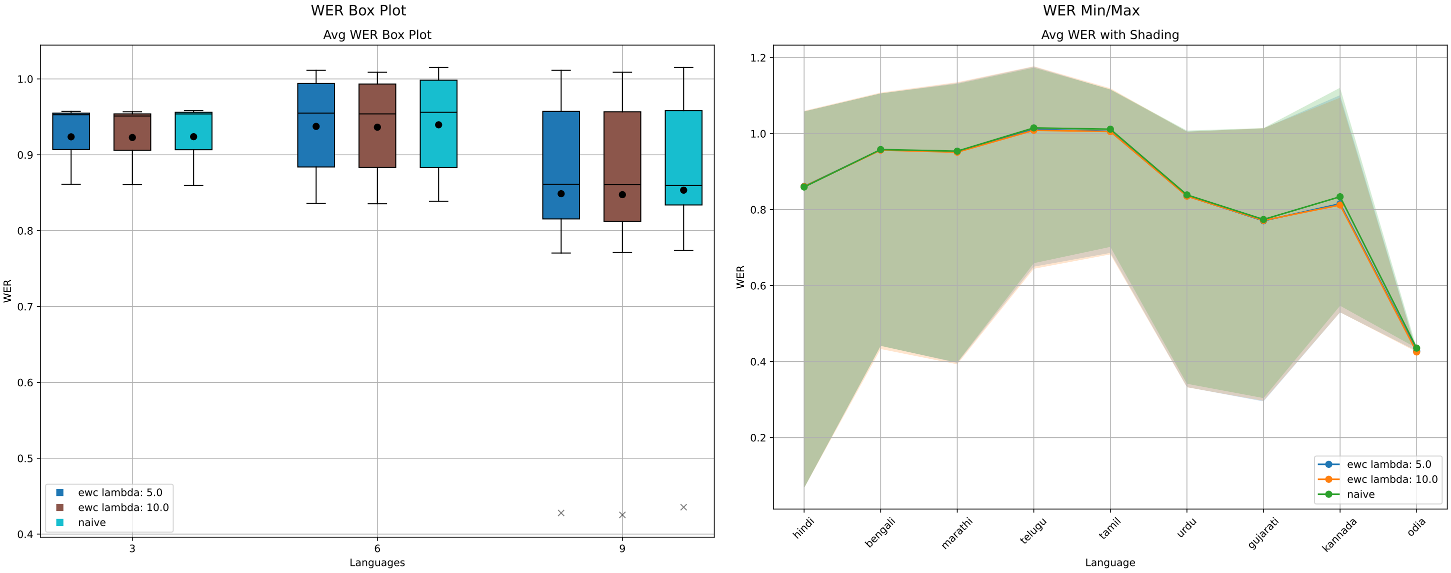


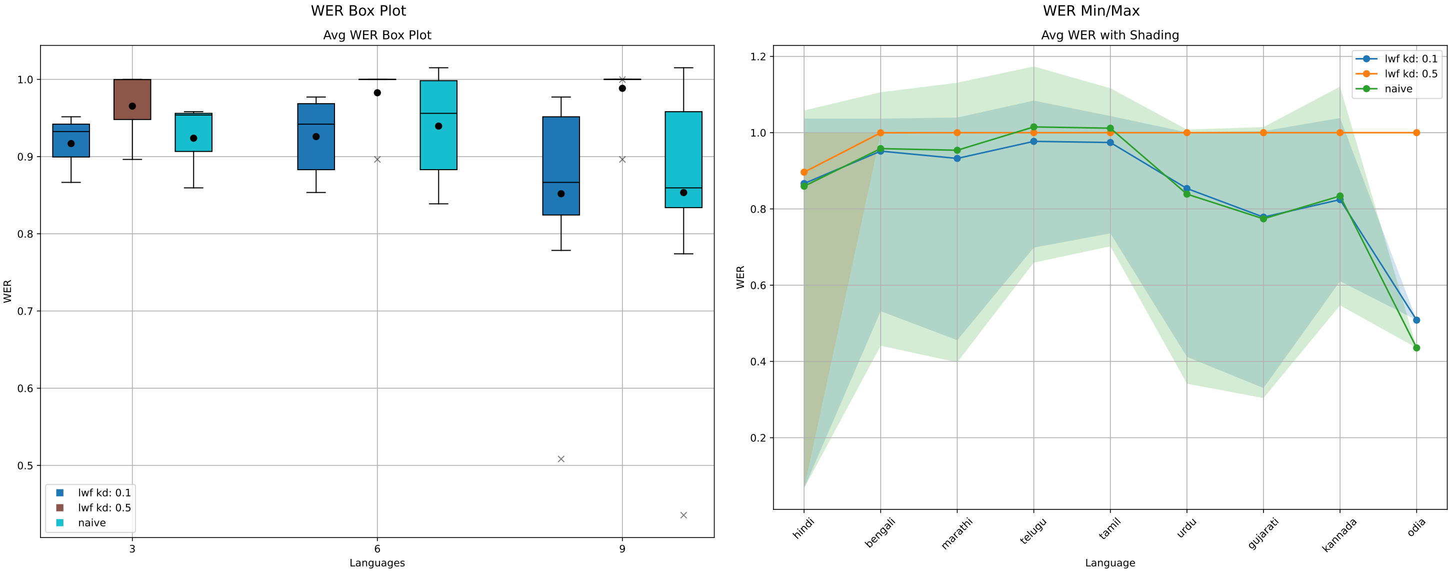
Figure 4: All comparison noisy shaded WER plot



(a) EWC Box

(b) EWC Shaded

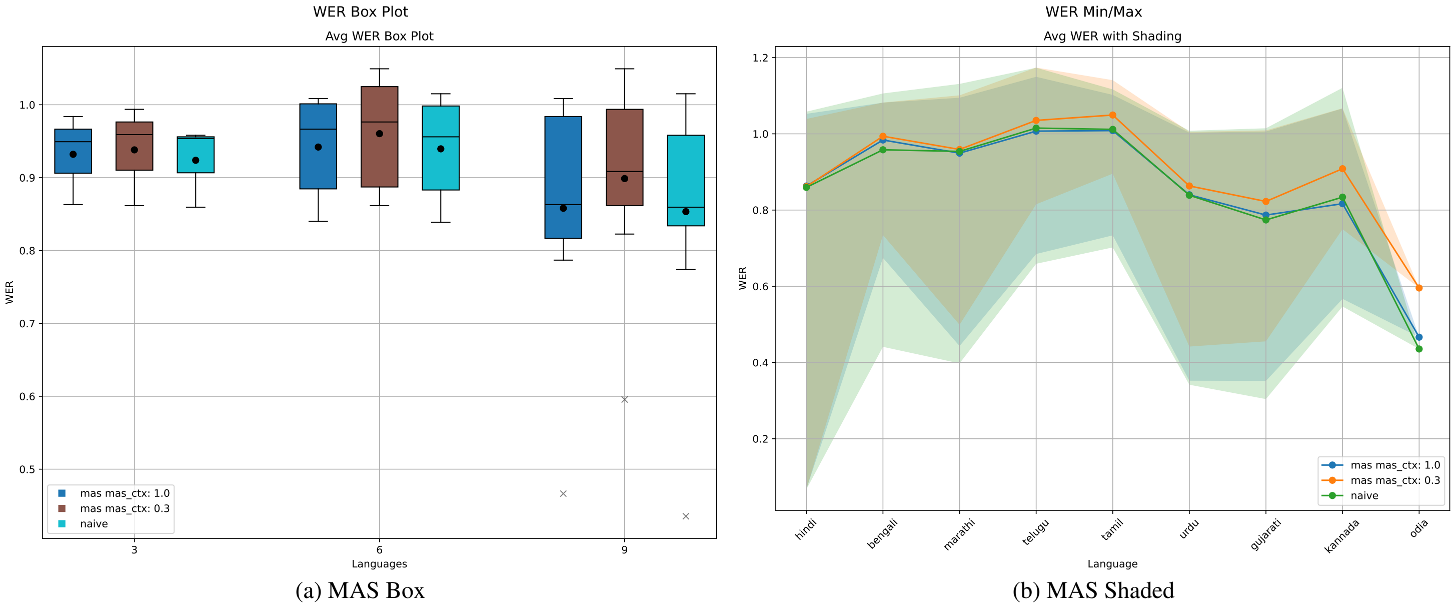
Figure 5: EWC Ablation – Box and Shaded Plots



(a) LWF Box

(b) LWF Shaded

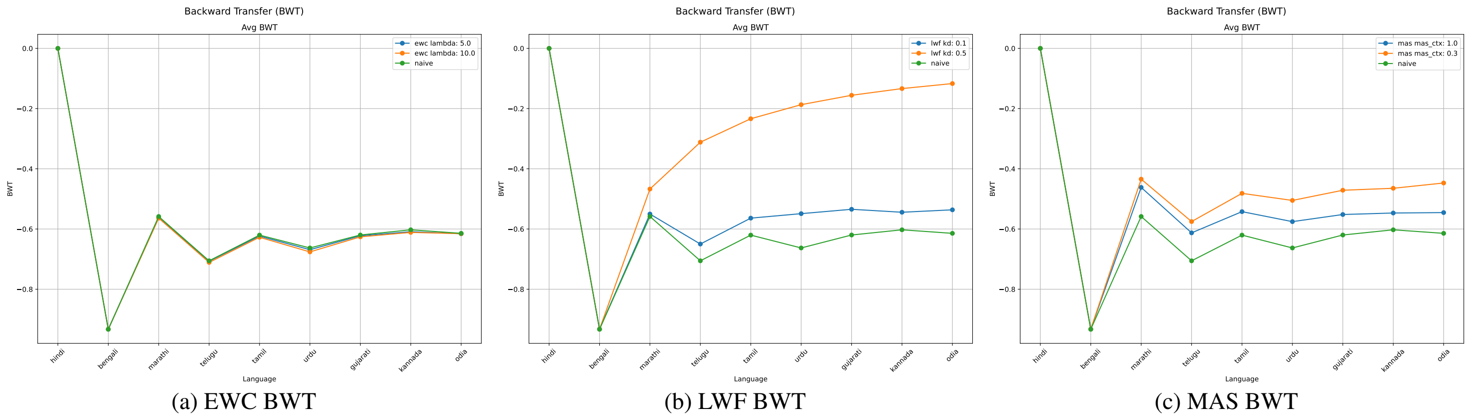
Figure 6: LWF Ablation – Box and Shaded Plots



(a) MAS Box

(b) MAS Shaded

Figure 7: MAS Ablation – Box and Shaded Plots

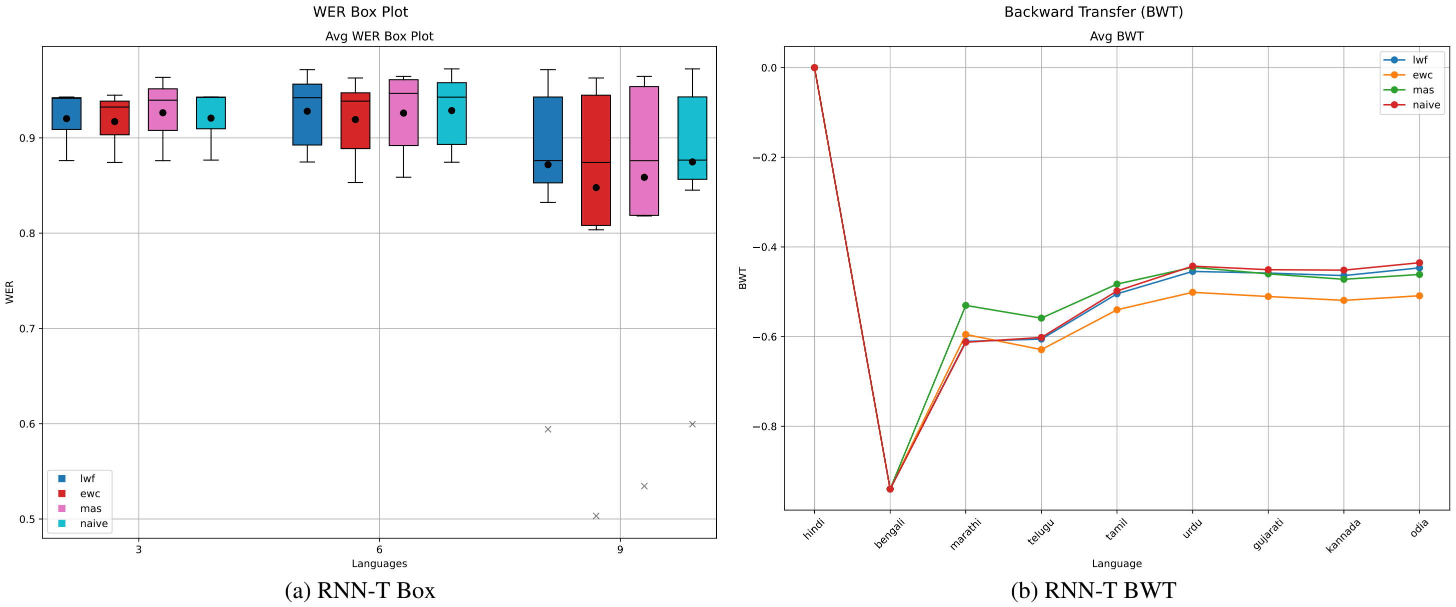


(a) EWC BWT

(b) LWF BWT

(c) MAS BWT

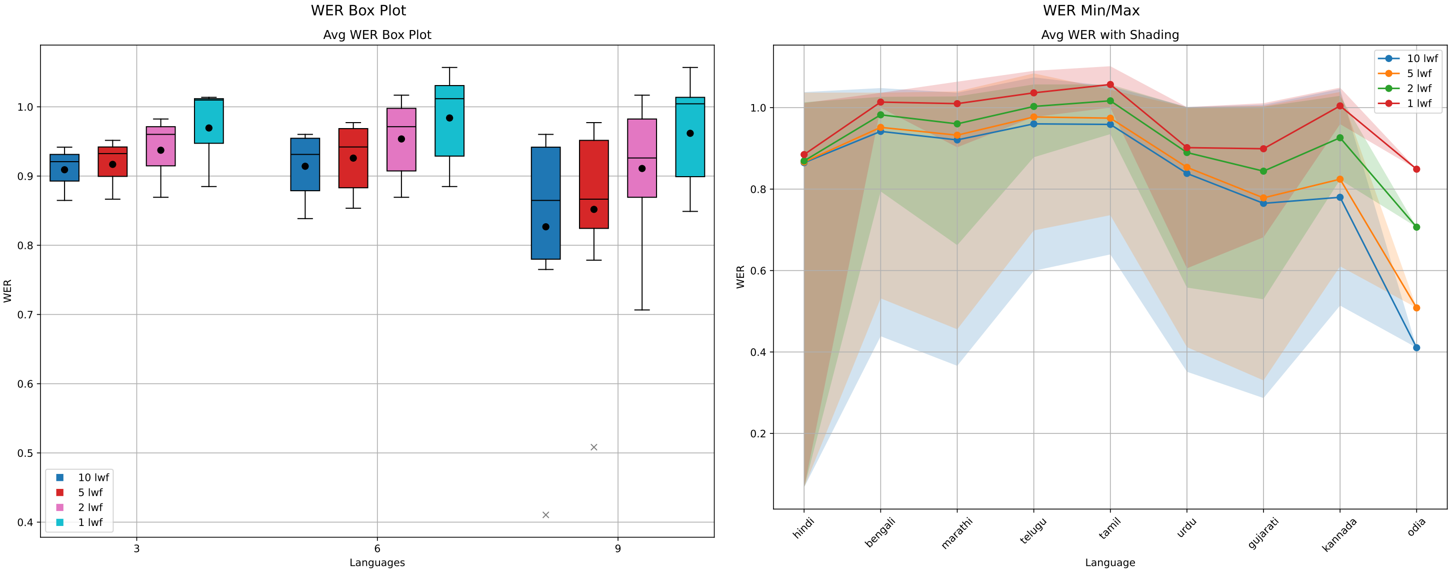
Figure 8: BWT Plots from EWC, LWF, and MAS Ablations



(a) RNN-T Box

(b) RNN-T BWT

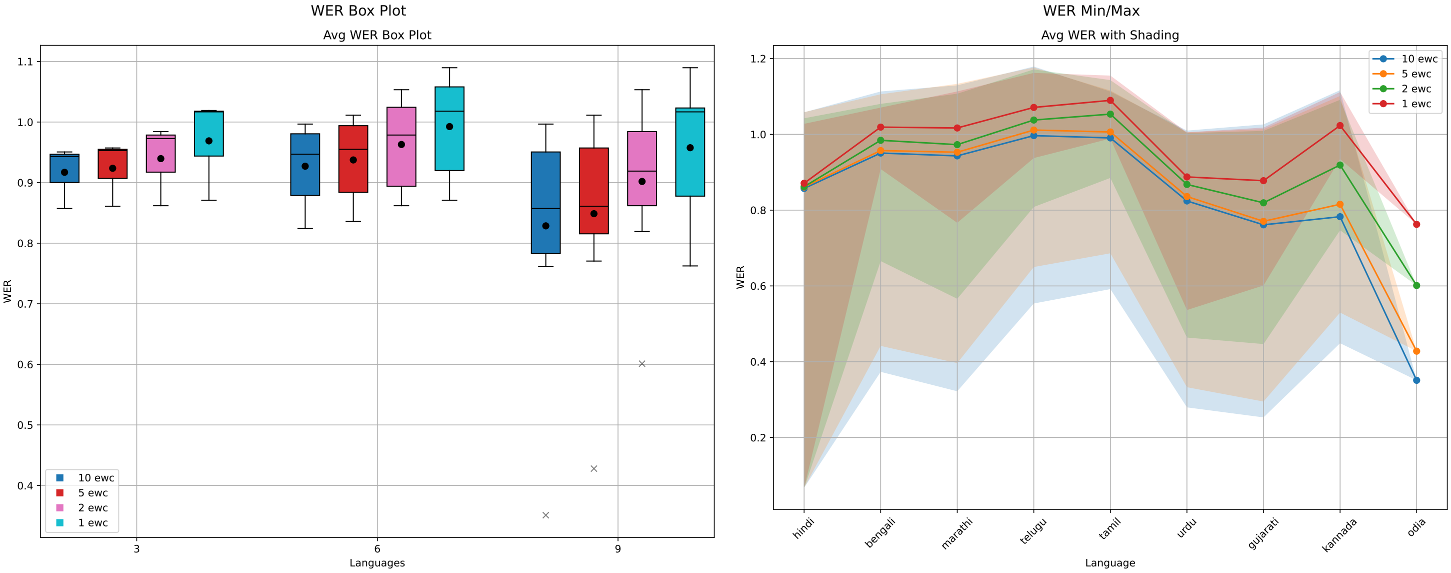
Figure 9: RNN-T Benchmark – Box and BWT Plots



(a) LWF Epoch Box

(b) LWF Epoch Shaded

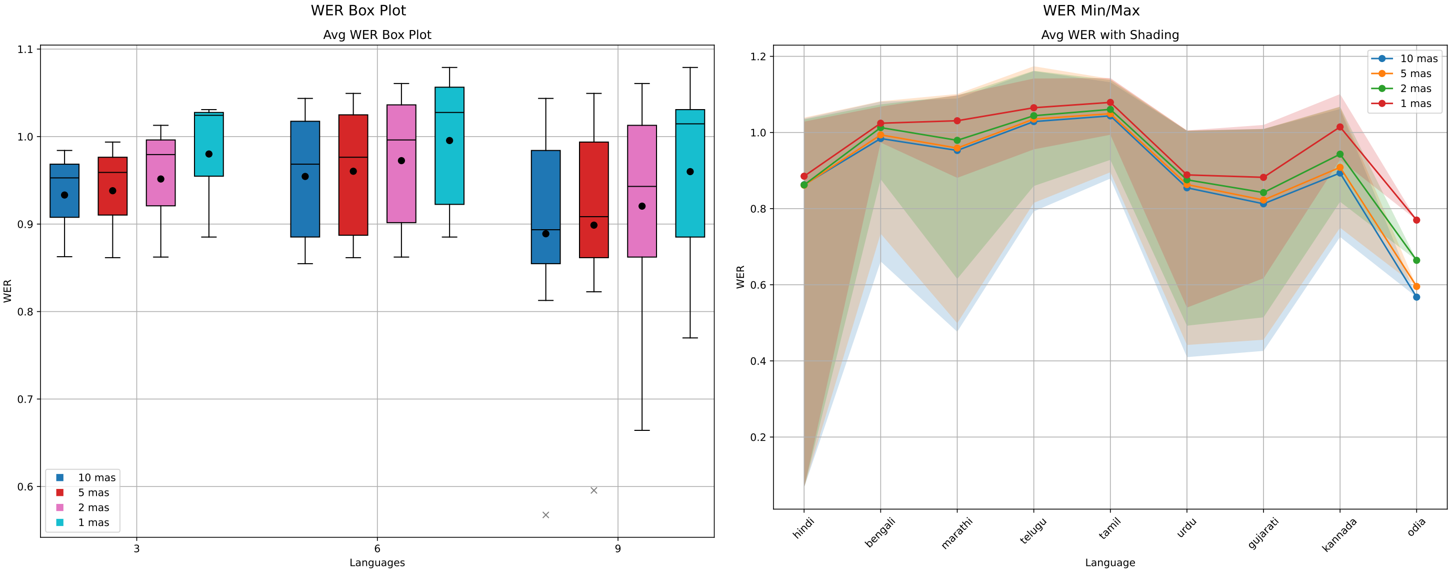
Figure 10: LWF Epoch – Box and Shaded Plots



(a) EWC Epoch Box

(b) EWC Epoch Shaded

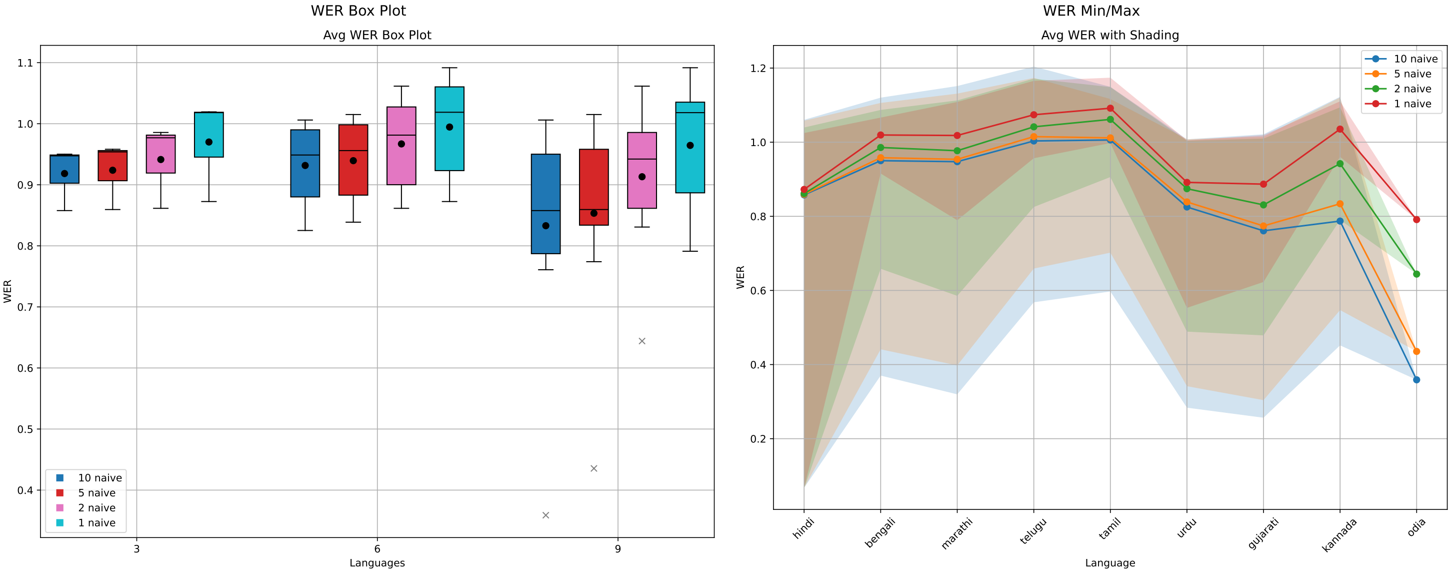
Figure 11: EWC Epoch – Box and Shaded Plots



(a) MAS Epoch Box

(b) MAS Epoch Shaded

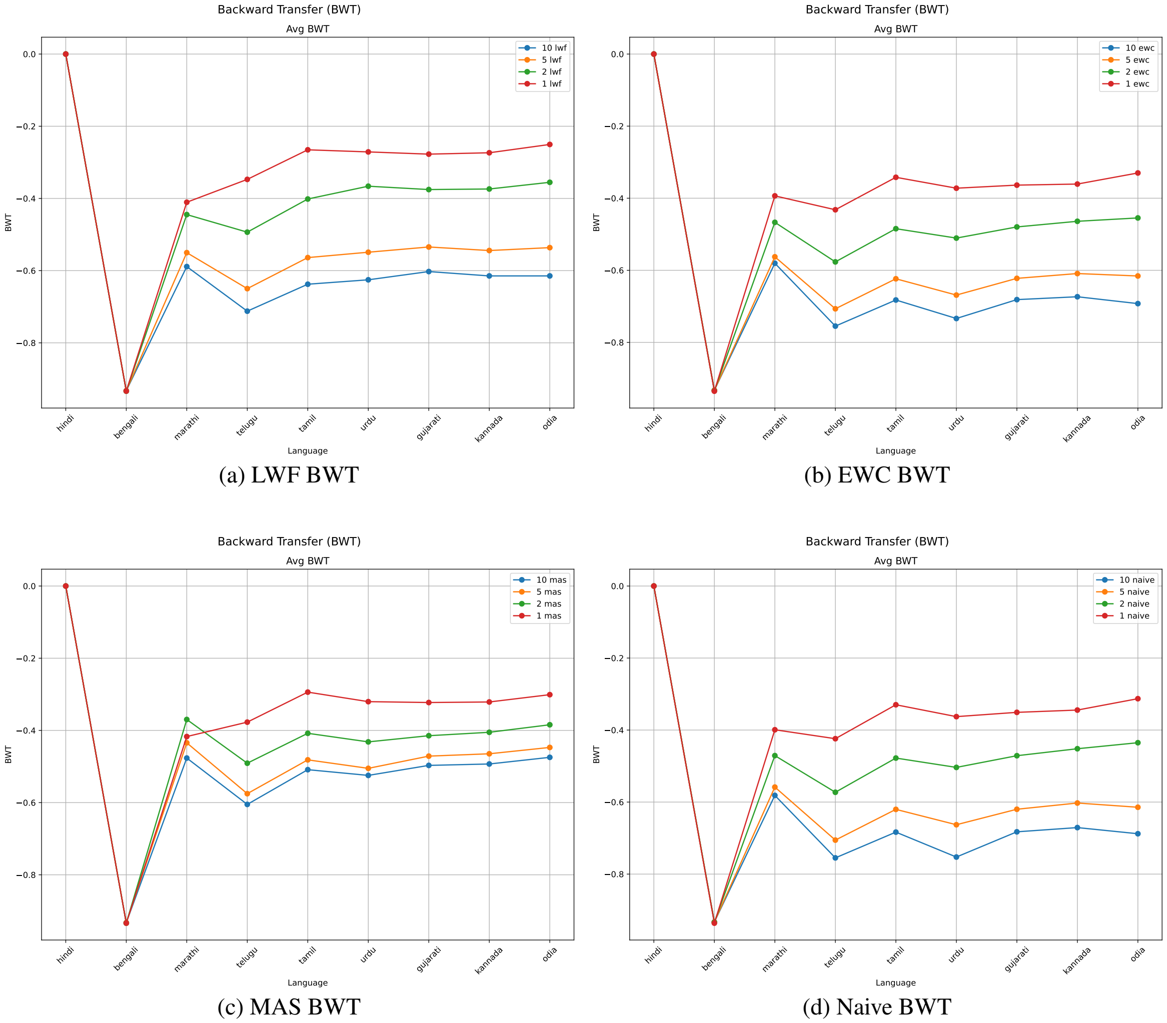
Figure 12: MAS Epoch – Box and Shaded Plots



(a) Naive Epoch Box

(b) Naive Epoch Shaded

Figure 13: Naive Epoch – Box and Shaded Plots



(a) LWF BWT

(b) EWC BWT

(c) MAS BWT

(d) Naive BWT

Figure 14: BWT Plots for Epoch-Wise Learning – LWF, EWC, MAS, and Naive