



Analyzing Mitigation Strategies for Catastrophic Forgetting in End-to-End Training of Spoken Language Models

Chi-Yuan Hsiao¹, Ke-Han Lu¹, Kai-Wei Chang¹, Chih-Kai Yang¹, Wei-Chih Chen¹, Hung-yi Lee¹

¹National Taiwan University, Taiwan

r12942086@ntu.edu.tw, dl2942024@ntu.edu.tw, kaiwei.chang.tw@gmail.com,
chihkaiyang1124@gmail.com, r12921120@ntu.edu.tw, hungyilee@ntu.edu.tw

Abstract

End-to-end training of Spoken Language Models (SLMs) commonly involves adapting pre-trained text-based Large Language Models (LLMs) to the speech modality through multi-stage training on diverse tasks such as ASR, TTS and spoken question answering (SQA). Although this multi-stage continual learning equips LLMs with both speech understanding and generation capabilities, the substantial differences in task and data distributions across stages can lead to catastrophic forgetting, where previously acquired knowledge is lost. This paper investigates catastrophic forgetting and evaluates three mitigation strategies—model merging, discounting the LoRA scaling factor, and experience replay to balance knowledge retention with new learning. Results show that experience replay is the most effective, with further gains achieved by combining it with other methods. These findings provide insights for developing more robust and efficient SLM training pipelines.

Index Terms: spoken language model, catastrophic forgetting, continual learning, model merging

1. Introduction

Inspired by the remarkable success of large language models (LLMs) [1, 2, 3, 4] in natural language processing (NLP), researchers have begun exploring *spoken language models* (SLMs)¹ as powerful solutions for speech processing tasks. For instance, textless SLMs [5] perform speech continuation without text supervision, while task-specific SLMs, such as VALL-E [6] for text-to-speech (TTS) and Seamless [7] for speech translation, leverage generative language modeling to achieve state-of-the-art performance. More recently, researchers have also investigated the instruction-following (IF) capability of SLMs, enabling them to tackle diverse speech processing tasks through natural language guidance. This advancement enhances the flexibility and adaptability of SLMs across various applications [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18].

Due to the high complexity of speech signals, advanced SLMs are typically built by incorporating pre-trained text LLMs rather than being trained from scratch. A common approach is to use a pre-trained text LLM as the backbone and adapt it to the speech modality, allowing it to understand and/or generate speech [13, 14, 15, 19]. These models can be broadly categorized based on how they incorporate speech into the LLM. One approach involves integrating a *speech encoder* with a text

¹Currently, there is no strict definition of SLMs. In this paper, we define SLMs as models that can process both text and speech as input and generate either text or speech as output, with at least one modality being speech. Without loss of generality, this paper focus on analysing SLMs capable of simultaneously accepting text and speech as input and producing both text and speech as output.

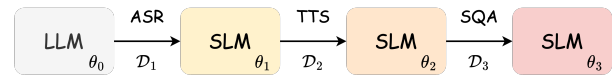


Figure 1: Continual training of a spoken language model using multi-stage speech processing tasks.

LLM through a projection network for representation alignment. The projection network are then trained optionally along with the LLM to make it familiar with the speech modality, as seen in models like Qwen-Audio [17, 18], SALMONN [14], and DeSTA[15, 16]. While these SLMs demonstrate strong speech understanding capabilities, they cannot generate speech responses.

Another approach directly integrates speech tokens (e.g., semantic tokens derived from self-supervised learning (SSL) speech models and acoustic tokens from speech codec models [20]) into the LLM, as seen in models [21, 22, 23, 24] such as Moshi and Mini-Omni. This usually requires *vocabulary expansion* [19, 22], where the LLM’s vocabulary is extended to include both text and speech tokens. By jointly modeling text and speech tokens, these SLMs can effectively understand and generate both modalities.

In order to familiarize LLMs with the speech modality, *multi-stage training* is often employed. This involves training the LLM on speech processing tasks across several stages, each using a distinct dataset, such as ASR, TTS, and Spoken Question Answering (SQA) as shown in Fig. 1. For example, during the ASR stage, the LLM is equipped with speech understanding capabilities, while in the TTS stage, the LLM learns to generate speech. In the SQA stage, the LLM gains the ability to answer questions in speech based on spoken input.

However, due to substantial differences in tasks and data distributions across stages, *catastrophic forgetting* [25] may occur, causing the LLM to lose previously acquired knowledge or abilities. Although SLMs gain new speech understanding capabilities, they must also retain their original text-based knowledge for tasks such as SQA or following speech instructions. Catastrophic forgetting can degrade the performance of SLMs in both text and speech modalities.

To study this problem and explore potential solutions, we examine three widely used strategies for mitigating catastrophic forgetting in LLMs and SLMs. These strategies include (1) model merging [26, 27], (2) discounting the LoRA scaling factor [14, 28], and (3) experience replay [29, 30, 31]. Specifically, in this paper, we train an SLM based on LLaMA[3] and systematically analyze catastrophic forgetting at each training stage by evaluating its performance on question answering and instruction-following tasks in the text modality to assess knowl-

edge retention. After applying mitigation strategies, we compare their effectiveness by evaluating SQA in the speech modality, along with the previously tested text-based tasks.

Overall, this study investigates catastrophic forgetting in SLM training and evaluates three commonly used mitigation strategies. Our experimental results show that among the three strategies examined, experience replay proves to be the most effective, significantly reducing knowledge loss while maintaining performance across both text and speech modalities.

2. Mitigation strategies

In this section, we present three common strategies for mitigating catastrophic forgetting in LLMs and SLMs, which are the focus of this paper: (1) model merging [26, 27], (2) discounting the LoRA scaling factor [32, 15], and (3) experience replay [29, 30, 31].

2.1. Model merging

Consider an SLM training process with N stages, where θ_i denotes the model weights after the i -th training stage. The complete set of model weights is given by $M = \{\theta_0, \theta_1, \theta_2, \dots, \theta_N\}$, where θ_0 represents the original pre-trained model, and θ_N the final SLM weights. To leverage information from multiple training stages, we explore model merging techniques that aggregate weights in M using several methods, including naive linear combination method, TIES [33], and DARE [34]. By applying these methods, we aim to preserve knowledge from different training stages, mitigating forgetting and enhancing the final performance.

2.2. Discounting LoRA-scaling factor

Given an input $\mathbf{x} \in \mathbb{R}^{d_{in}}$ and a weight matrix $\mathbf{W} \in \mathbb{R}^{d_{out} \times d_{in}}$, the forward pass with a LoRA adapter of rank r is given by:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \frac{\alpha}{r}(\mathbf{B}\mathbf{A}\mathbf{x}), \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{r \times d_{in}}$ and $\mathbf{B} \in \mathbb{R}^{d_{out} \times r}$ are the weight of the adapter, and α is the scaling factor. The strategy is to set a lower α of model that is finetuned with LoRA adapter during inference process to make the effect of adapter weaker.

2.3. Experience replay

Different from above strategies, experience replay is applied while model training. Suppose a pre-trained model θ_0 is initially trained on a dataset \mathcal{D}_0 and will undergo N additional training stages. We define the set of training datasets as:

$$D = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N\}, \quad (2)$$

where \mathcal{D}_i represents the dataset used in the i -th training stage.

At each stage i , experience replay is applied to construct an augmented dataset \mathcal{D}'_i by including random samples from all previous datasets as well as \mathcal{D}_0 , defined as:

$$\mathcal{D}'_i = \mathcal{D}_i \cup \bigcup_{j=0}^{i-1} \text{Sample}(\mathcal{D}_j, s|\mathcal{D}_i|), \quad (3)$$

where $\text{Sample}(\mathcal{D}, k)$ means randomly sample k examples from dataset \mathcal{D} . s is the sampling ratio, determining the proportion of each previous dataset \mathcal{D}_j included in \mathcal{D}'_i . Since each dataset \mathcal{D}_i may corresponds to different task, the multi-task

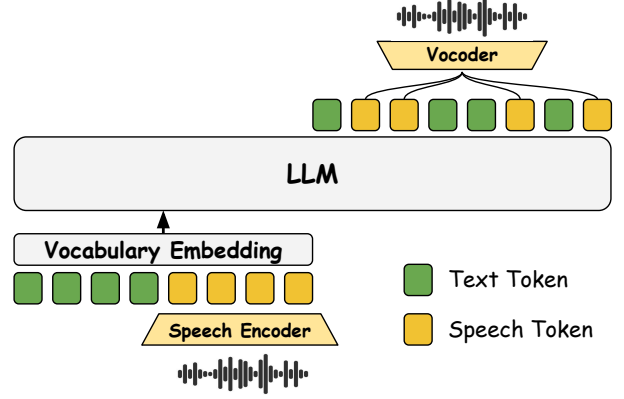


Figure 2: The architecture of a Spoken Language Model (SLM), which consists of a backbone LLM, a speech encoder that converts speech into speech tokens, and a vocoder that synthesizes the speech tokens into a speech waveform.

learning is applied when training with the augmented dataset \mathcal{D}'_i . Notably, the number of random samples is according to the size of i -th dataset $|\mathcal{D}_i|$.

3. Experimental setup

3.1. Spoken language model

3.1.1. Model architecture

As shown in Figure 2, our SLM comprises three main components: a speech encoder, a LLM backbone, and a vocoder. The speech encoder extracts speech features from speech waveforms, subsequently quantized into discrete speech tokens via k-means clustering. These tokens are incorporated into the LLM’s vocabulary for language modeling. Finally, the vocoder reconstructs speech waveform from the generated speech tokens.

3.1.2. Training methods

We fine-tune the LLM in three stages of instruction-tuning for different tasks, including automatic speech recognition (ASR), text-to-speech synthesis (TTS), and spoken question answering (SQA). During fine-tuning, loss is computed only on the model’s response. Formally, let \mathbf{P} denotes a prompt and \mathbf{R} the model’s response. A text sentence \mathbf{T} with L words is defined as $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \dots, \mathbf{t}_L]$, where \mathbf{t}_i represents the text token sequence of the i -th word. Similarly, a speech utterance \mathbf{S} with L spoken words is defined as $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_L]$, where \mathbf{s}_i represents the speech token sequence of the i -th word. Each \mathbf{t}_i and \mathbf{s}_i can have varying lengths, containing a different number of text tokens and speech tokens, respectively. We outline the data formulation and methodology for each training stage:

ASR stage: To align speech and text tokens, we first train the model on automatic speech recognition (ASR). In this stage, the model learns to generate a text transcription \mathbf{T}_{ASR} given a text instruction \mathbf{T}_I for ASR and a speech utterance \mathbf{S}_{ASR} . The prompt \mathbf{P} and the model response \mathbf{R} are shown as:

$$\mathbf{P} = [\mathbf{T}_I, \mathbf{S}_{ASR}], \mathbf{R} = [\mathbf{T}_{ASR}], \quad (4)$$

where \mathbf{S}_{ASR} and \mathbf{T}_{ASR} are the speech-text pair in ASR dataset.

TTS stage: Next, the model learns speech generation using the same ASR dataset, as speech-text pairs are also required

for text-to-speech synthesis (TTS). In this stage, the model generates text and speech tokens in an alternating, word-by-word interleaved manner—an approach crucial for successful training, as the model often struggles to converge without it. Given a text instruction \mathbf{T}_I for TTS and a text sentence \mathbf{T}_{ASR} . The prompt and model response are:

$$\mathbf{P} = [\mathbf{T}_I, \mathbf{T}_{ASR}], \mathbf{R} = [\mathbf{t}_1, \mathbf{s}_1, \mathbf{t}_2, \mathbf{s}_2, \dots, \mathbf{t}_L, \mathbf{s}_L], \quad (5)$$

where $\mathbf{t}_i \in \mathbf{T}_{ASR}, \mathbf{s}_i \in \mathbf{S}_{ASR} \forall i = 1, 2, \dots, L$, and \mathbf{S}_{ASR} and \mathbf{T}_{ASR} are from the same speech-text pairs in ASR dataset.

SQA stage: Finally, the model learns spoken question answering (SQA) by leveraging its ASR and TTS capabilities. Given a spoken question \mathbf{S}_Q , the model first predicts its text transcription \mathbf{T}_Q , followed by the text answer \mathbf{T}_A and its word-by-word interleaved text-speech representation. The prompt and response are structured as:

$$\mathbf{P} = [\mathbf{S}_Q], \mathbf{R} = [\mathbf{T}_Q, \mathbf{T}_A, \mathbf{t}_1, \mathbf{s}_1, \mathbf{t}_2, \mathbf{s}_2, \dots, \mathbf{t}_L, \mathbf{s}_L], \quad (6)$$

where $\mathbf{t}_i \in \mathbf{T}_A, \mathbf{s}_i \in \mathbf{S}_A \forall i = 1, 2, \dots, L$. $\mathbf{S}_Q, \mathbf{T}_Q, \mathbf{S}_A$ and \mathbf{T}_A originate from the same SQA dataset example.

Experience replay: During training with experience replay, data formulation follows the same structure as in each training stage, including samples from both the current and previous stages. The initial dataset \mathcal{D}_0 , representing the LLM’s original training data, is assumed to contain text instruction-response pairs. If such a dataset is available, the prompt and response are formulated as:

$$\mathbf{P} = [\mathbf{T}_I], \mathbf{R} = [\mathbf{T}_R], \quad (7)$$

where \mathbf{T}_I and \mathbf{T}_R are the text instruction-response pair in \mathcal{D}_0 .

3.1.3. Training details

We follow SeamlessM4T v2’s settings [7] for speech token extraction and reconstruction. Our model uses xlsr2-1b-v2 [35] as speech encoder with k-means clustering [36] to obtain 10,000 discrete speech tokens. The LLM is based on Llama-3.2-11B-Vision-Instruct, excluding the vision encoder. For vocoding, we adopt the pre-trained HiFi-GAN [37] from SeamlessM4T v2, supporting multiple speakers and languages. Our hyperparameters include LoRA adapters of rank $r = 64$ and $\alpha = 16$ for self-attention matrices, with full fine-tuning of the embedding layer, language model heads, and the last five self-attention layers. We optimize the model using the AdamW optimizer with a learning rate of $1e-5$ and a warmup ratio 0.1. In ASR and TTS stage, we train the whole dataset for 2 epochs with batch size 4 and set max sequence length to 800. In SQA stage, we train the whole dataset for 1 epoch with batch size 1 and set max sequence length to 1200. All experiments were conducted on four Nvidia RTX A6000 GPUs, with the full training process taking approximately five days.

3.2. Mitigation strategies

3.2.1. Model merging

Our training process consists of three stages, resulting in a set of model weights denoted as $M = \{\theta_0, \theta_1, \theta_2, \theta_3\}$, where $\theta_0, \theta_1, \theta_2$, and θ_3 represent the model weights at the initial stage, after the ASR stage, after the TTS stage, and after the SQA stage, respectively. The following introduces the settings for each model merging method:

Linear Combination: weight = [0.02, 0.03, 0.05, 0.9]

TIES: weight = [−, 0.04, 0.06, 0.9], density = [−, 0.9, 0.9, 0.9], BaseModel = θ_0

DARE: weight = [−, 0.04, 0.06, 0.9], density = [−, 0.9, 0.9, 0.9], BaseModel = θ_0

Discounting LoRA-scaling factor Restricted to computation budget, we only choose $\alpha = 15$ and $\alpha = 14$ as hyperparameter of LoRA adapters for evaluation.

Experience replay: We evaluate two training settings: with and without experience replay across all training stages. When applying experience replay, we use a subsampling ratio of $s = 0.005$. For the initial dataset \mathcal{D}_0 , the LLM’s original training data, we select a text instruction-tuning dataset generated by the same LLM. We assume that its distribution closely approximates that of the original training data.

Mixed strategy: In our evaluation, we also apply additional mitigation strategies to models trained with experience replay, resulting in two additional baseline settings: “model merging after experience replay” and “discounting the LoRA scaling factor after experience replay”. All mitigation strategy parameters remain consistent with the previous settings.

3.3. Datasets

This section describes datasets, benchmarks, and their preprocessing methods used for training and evaluation.

3.3.1. Training

We use LibriSpeech 960-hour[38] for ASR and TTS training and Magpie-Air [39]² for SQA and experience replay.

In ASR stage, LibriSpeech is used for training, with each example assigned one of ten randomly selected instructions, such as “Please repeat the following words:”.

In TTS stage, LibriSpeech is also used, with randomly assigned instructions like “Please speak out loud the following words:”. To generate word-by-word text-speech interleaved sequences, we align text transcriptions and speech utterances at the word level using Whisper-Timestamped³.

In SQA stage, Magpie-Air is used for training. We filter examples based on length and topic, removing categories like math and coding that are challenging to describe in speech. Speech questions and answers are synthesized from text using SpeechT5 [40], creating both text and speech versions of QA pairs. Whisper-Timestamped is used to align answers and generate interleaved text-speech token sequences.

When applying experience replay, we use Magpie-Air as \mathcal{D}_0 , randomly sampling text instruction-response pairs for training. Since Magpie-Air is constructed by prompting Llama 3 8B, an LLM from the same series as ours, it serves as a suitable dataset for experience replay.

3.3.2. Evaluation

To assess the model’s learned capabilities, we evaluate spoken question answering (SQA) on the speech modality. To measure the catastrophic forgetting in the SLM, we evaluate question answering (QA) and instruction-following on the text modality.

Spoken question answering: Following the evaluation settings of Moshi and Spectron [41], we use three datasets for SQA: (1) Spoken WebQuestions, (2) LLaMA-Questions and (3) Audio Trivia QA. SQA is evaluated under two settings: speech-to-text (S2T), where accuracy is computed by directly matching

²Magpie-Align/Llama-3-Magpie-Air-3M-v0.1

³<https://github.com/linto-ai/whisper-timestamped>

Table 1: Accuracies (%) for various strategies to mitigate catastrophic forgetting. **Merge**: Model merging. **Scaling**: Scaling the LoRA factor. **w/ R**: With Experience Replay. The results are reported across four datasets: LLaMA, Web, Trivia, and IFEval, with each dataset further evaluated on different tasks (T2T, S2T, S2S).

Mitigation Strategy	LLaMA			Web			Trivia			IFEval	
	T2T	S2T	S2S	T2T	S2T	S2S	T2T	S2T	S2S	T2T Prompt	T2T Instruction
Original	70.0	-	-	61.5	-	-	78.9	-	-	67.1	77.1
None	14.3	7.3	8.0	3.6	1.5	0.8	6.0	3.1	1.9	9.2	20.1
Merge (Linear)	19.3	9.0	7.7	5.6	1.1	0.6	7.8	3.9	1.1	8.5	19.3
Merge (TIES)	12.7	6.3	2.7	3.7	0.7	0.0	4.9	2.0	0.4	10.9	22.2
Merge (DARE)	4.0	6.0	1.7	1.2	0.7	0.0	1.3	2.0	0.3	11.8	23.1
Scaling ($\alpha = 15$)	15.0	7.0	6.7	4.5	1.3	1.1	5.9	3.9	2.3	9.1	18.6
Scaling ($\alpha = 14$)	16.0	7.3	6.7	4.0	1.2	0.3	0.3	3.3	1.9	7.9	18.3
Replay	66.3	50.3	28.7	55.2	24.2	9.1	66.4	25.2	11.1	47.5	57.9
Merge (Linear) w/ R	68.0	44.7	16.7	56.4	19.2	3.6	68.8	16.3	5.2	50.3	61.3
Merge (TIES) w/ R	66.7	42.0	17.0	55.3	17.6	2.7	67.5	15.9	4.8	50.1	60.0
Merge (DARE) w/ R	69.3	40.0	15.7	53.1	18.4	3.0	64.0	14.3	3.8	43.8	52.2
Scaling ($\alpha = 15$) w/ R	68.7	52.7	28.7	54.4	24.9	8.0	68.6	25.4	11.2	43.7	59.4
Scaling ($\alpha = 14$) w/ R	68.7	50.7	28.7	55.3	25.5	6.9	66.5	27.2	11.1	49.0	60.6

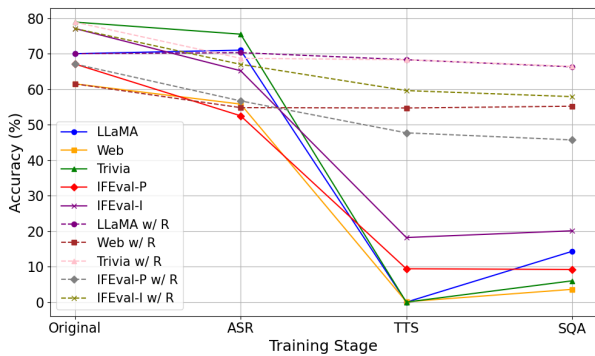


Figure 3: Evaluation results on instruction-following and question answering. LLaMA, Web, and Trivia denote LLaMA-Questions, Spoken WebQuestions, and Audio Trivia QA. IFEval-P and IFEval-I stand for IFEval in prompt-level and instruction-level. w/ R means with experience replay.

text responses, and speech-to-speech (S2S), where the speech responses is first transcribed with Whisper-large-v3 and considered correct if the transcription contains the correct answer.

Question answering: We use the same datasets as in SQA but evaluate only on the text modality (Text-to-Text, T2T). Accuracy is used as the evaluation metric.

Instruction following: We use IFEval [42] for evaluating instruction-following on the text modality (T2T setting). IFEval computes accuracy at both the prompt and instruction level to assess the model’s ability to follow instructions accurately.

4. Results

4.1. Catastrophic forgetting

Fig.3 shows the evaluation results on instruction-following and question answering in each training stage on T2T setting. For SLM without any mitigation strategy, it is obvious that catastrophic forgetting appear during training. As training stage moves on, the accuracy of both evaluation tasks decrease in different degrees. There is a gap can be observed easily between ASR stage and TTS stage on both evaluation tasks, which also

shows that the most serious forgetting appears in TTS stage. Interestingly, accuracy for question answering slightly grow in SQA stage. We assume this is because the SLM recalls some of its knowledge from the text sequence at this stage, even if only to a small extent. As for SLM applied with experience replay, although there is still a little forgetting during training, the extent has been mitigated significantly compared to one without experience replay.

4.2. Mitigation strategies

Table.1 shows the results for all mitigation strategies. From the results, the are several findings:

(1) **Experience replay surpasses the other strategies:** According to the results, experience replay surpasses all the other single strategies on tasks evaluated for mitigation (T2T) as well as new ability (S2T, S2S).

(2) **Mixed Strategy can further boost the performance:** Compared to experience replay, other mixed strategies can achieve better performance of new ability in S2T setting in some cases. However, mixed strategy with discounting LoRA-scaling factor more robust than model merging.

(3) **Experience replay remains robustness in every setting:** Although all mitigation strategies can do some improvements in almost all settings. However, only experience replay remains robust in S2S setting and surpasses other strategies.

5. Conclusion

This paper investigates mitigation strategies for continual learning in developing spoken language models (SLMs) from large language models (LLMs). The results demonstrate that experience replay is the most effective method, with further performance gains achievable by combining it with other techniques. Through a case study, we highlight catastrophic forgetting as a significant challenge and showcase the potential of these strategies to address it. Future work will involve more comprehensive studies, including diverse training pipelines, models and various strategies to inspire the speech community to develop more efficient training methods.

6. References

- [1] OpenAI, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [2] G. Team *et al.*, “Gemini: a family of highly capable multimodal models,” *arXiv preprint arXiv:2312.11805*, 2023.
- [3] A. Dubey *et al.*, “The llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.
- [4] A. Yang *et al.*, “Qwen2.5 technical report,” *arXiv preprint arXiv:2412.15115*, 2024.
- [5] K. Lakhota *et al.*, “On generative spoken language modeling from raw audio,” vol. 9, pp. 1336–1354, 2021.
- [6] S. Chen *et al.*, “Neural codec language models are zero-shot text to speech synthesizers,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 33, pp. 705–718, 2025.
- [7] L. Barrault *et al.*, “Seamless: Multilingual expressive and streaming speech translation,” *arXiv preprint arXiv:2312.05187*, 2023.
- [8] C. yu Huang *et al.*, “Dynamic-superb phase-2: A collaboratively expanding benchmark for measuring the capabilities of spoken language models with 180 tasks,” 2024. [Online]. Available: <https://arxiv.org/abs/2411.05361>
- [9] C.-Y. Huang *et al.*, “Dynamic-superb: Towards a dynamic, collaborative, and comprehensive instruction-tuning benchmark for speech,” in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 12 136–12 140.
- [10] S. Arora, K.-W. Chang *et al.*, “On the landscape of spoken language models: A comprehensive survey,” *arXiv preprint arXiv:2504.08528*, 2025.
- [11] S. Arora *et al.*, “UniverSLU: Universal spoken language understanding for diverse tasks with natural language instructions,” in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, K. Duh, H. Gomez, and S. Bethard, Eds., Jun. 2024, pp. 2754–2774.
- [12] J. Tian *et al.*, “ESPnet-SpeechLM: An open speech language model toolkit,” in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, Apr. 2025, pp. 116–124.
- [13] Y. Gong *et al.*, “Joint audio and speech understanding,” in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023.
- [14] C. Tang *et al.*, “SALMONN: Towards generic hearing abilities for large language models,” in *The Twelfth International Conference on Learning Representations*, 2024.
- [15] K.-H. Lu *et al.*, “Desta: Enhancing speech language models through descriptive speech-text alignment,” in *Proc. Interspeech 2024*, 2024, pp. 4159–4163.
- [16] K.-H. Lu, Z. Chen, S.-W. Fu, C.-H. H. Yang, J. Balam, B. Ginsburg, Y.-C. F. Wang, and H.-y. Lee, “Developing instruction-following speech language model without speech instruction-tuning data,” *arXiv preprint arXiv:2409.20007*, 2024.
- [17] Y. Chu *et al.*, “Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models,” *arXiv preprint arXiv:2311.07919*, 2023.
- [18] Y. Chu, J. Xu, Q. Yang, H. Wei, X. Wei, Z. Guo, Y. Leng, Y. Lv, J. He, J. Lin *et al.*, “Qwen2-audio technical report,” *arXiv preprint arXiv:2407.10759*, 2024.
- [19] D. Zhang *et al.*, “SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*, Dec. 2023, pp. 15 757–15 773.
- [20] Z. Borsos *et al.*, “Audiolm: A language modeling approach to audio generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2523–2533, 2023.
- [21] A. Défossez *et al.*, “Moshi: a speech-text foundation model for real-time dialogue,” *arXiv preprint arXiv:2410.00037*, 2024.
- [22] Z. Xie and C. Wu, “Mini-omni: Language models can hear, talk while thinking in streaming,” *arXiv preprint arXiv:2408.16725*, 2024.
- [23] A. Zeng *et al.*, “Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot,” *arXiv preprint arXiv:2412.02612*, 2024.
- [24] C.-K. Yang *et al.*, “Building a taiwanese mandarin spoken language model: A first attempt,” *arXiv preprint arXiv:2411.07111*, 2024.
- [25] I. J. Goodfellow *et al.*, “An empirical investigation of catastrophic forgetting in gradient-based neural networks,” *arXiv preprint arXiv:1312.6211*, 2013.
- [26] E. Yang *et al.*, “Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities,” *arXiv preprint arXiv:2408.07666*, 2024.
- [27] Y. Lin *et al.*, “Mitigating the alignment tax of rlhf,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 580–606.
- [28] K.-H. Lu *et al.*, “Desta: Enhancing speech language models through descriptive speech-text alignment,” in *Interspeech 2024*, 2024, pp. 4159–4163.
- [29] D. Rolnick *et al.*, “Experience replay for continual learning,” *Advances in neural information processing systems*, vol. 32, 2019.
- [30] J. Zheng *et al.*, “Lifelong learning of large language model based agents: A roadmap,” *arXiv preprint arXiv:2501.07278*, 2025.
- [31] X. Zhang *et al.*, “Vqacl: A novel visual question answering continual learning setting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 102–19 112.
- [32] C. Tang *et al.*, “Salmonn: Towards generic hearing abilities for large language models,” in *The Twelfth International Conference on Learning Representations*.
- [33] P. Yadav, D. Tam *et al.*, “Ties-merging: Resolving interference when merging models,” in *Advances in Neural Information Processing Systems*, A. Oh *et al.*, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 7093–7115.
- [34] L. Yu *et al.*, “Language models are super mario: Absorbing abilities from homologous models as a free lunch,” in *Forty-first International Conference on Machine Learning*.
- [35] A. Conneau *et al.*, “Unsupervised cross-lingual representation learning for speech recognition,” in *Interspeech 2021*, 2021, pp. 2426–2430.
- [36] K. Krishna and M. Narasimha Murty, “Genetic k-means algorithm,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433–439, 1999.
- [37] J. Kong *et al.*, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” *Advances in neural information processing systems*, vol. 33, pp. 17 022–17 033, 2020.
- [38] V. Panayotov *et al.*, “Librispeech: An asr corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.
- [39] Z. Xu *et al.*, “Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing,” *arXiv preprint arXiv:2406.08464*, 2024.
- [40] J. Ao and R. a. Wang, “SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, May 2022, pp. 5723–5738.
- [41] E. Nachmani *et al.*, “Spoken question answering and speech continuation using spectrogram-powered llm,” *arXiv preprint arXiv:2305.15255*, 2023.
- [42] J. Zhou *et al.*, “Instruction-following evaluation for large language models,” *arXiv preprint arXiv:2311.07911*, 2023.