

SPADE: STRUCTURED PRUNING AND ADAPTIVE DISTILLATION FOR EFFICIENT LLM-TTS

Tan Dat Nguyen¹, Jaehun Kim¹, Ji-Hoon Kim¹, Shukjae Choi², Youshin Lim², Joon Son Chung¹

¹Korea Advanced Institute of Science and Technology, South Korea, ²42dot Inc., South Korea

ABSTRACT

The goal of this paper is to introduce SPADE, a framework for Structured Pruning and Adaptive Distillation for Efficient Large Language Model-based text-to-speech (LLM-TTS). Recent LLM-TTS systems achieve strong controllability and zero-shot generalization, but their large parameter counts and high latency limit real-world deployment. SPADE addresses this by combining (i) a *pruning* step guided by a word-error-rate-based layer importance index to remove non-essential Transformer layers, with (ii) multi-level *knowledge distillation* to restore autoregressive coherence. On zero-shot benchmarks, SPADE preserves near-parity perceptual quality while halving Transformer depth, reducing VRAM usage by up to 20%, and achieving up to $1.7\times$ faster real-time factor with less than 5% of the original training data. These results show that compact LLM-TTS models can maintain naturalness and speaker similarity while enabling practical real-time speech generation. Audio samples are available at ¹.

Index Terms— text-to-speech, LLM-TTS, knowledge distillation, pruning, speech synthesis

1. INTRODUCTION

Large Language Model (LLM)-based text-to-speech (LLM-TTS) systems such as CosyVoice [1, 2], VALL-E [3], CLaM-TTS [4], RALL-E [5], and LLaSA [6] have shown advanced controllability, prosody modeling, and zero-shot generalization across speakers and languages. Early approaches trained LLM backbones directly on speech tokens [1, 7], while recent methods initialize from pretrained text LLMs (e.g., LLaMA [8], Qwen2.5 [9]) and adapt them with speech objectives [6, 10]. Leveraging rich contextual representations, these systems synthesize natural speech conditioned on long prompts, speaker embeddings, and control tokens, pushing TTS closer to human-level performance.

Despite these advances, LLM-TTS models inherit the costly nature of text-only LLMs, including large

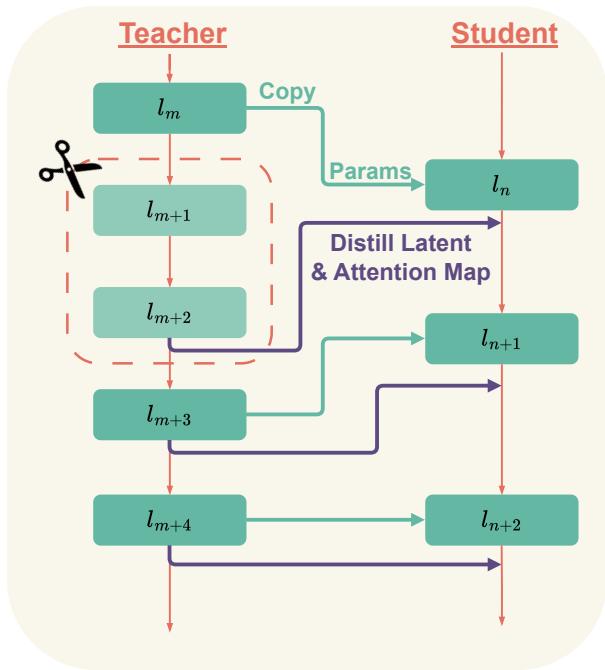


Fig. 1: Overview of SPADE. A large LLM-TTS model is compressed into a smaller student model through *pruning* and multi-level *distillation*. Parameters are copied from retained layers, while latent states are aligned across pruned segments to preserve synthesis quality.

parameter counts, high memory usage, and slow autoregressive decoding, and these factors are particularly pronounced in real-time deployment and on-device applications. On the other hand, a line of research in text-only LLM domain has extensively studied model compression, including pruning [11, 12, 13], distillation [14, 15, 16], quantization [17, 18, 19], and adaptive inference methods such as early exiting and token reduction [20, 21]. However, systematic compression methodology for LLM-TTS, where the preservation of prosody, naturalness, and long-context coherence serve as additional key aspects, remains underexplored.

In this paper, we present **SPADE**, a framework for **Structured Pruning** and **Adaptive Distillation** for Efficient LLM-TTS. By removing non-essential layers

¹<https://mm.kaist.ac.kr/projects/SPADE/>

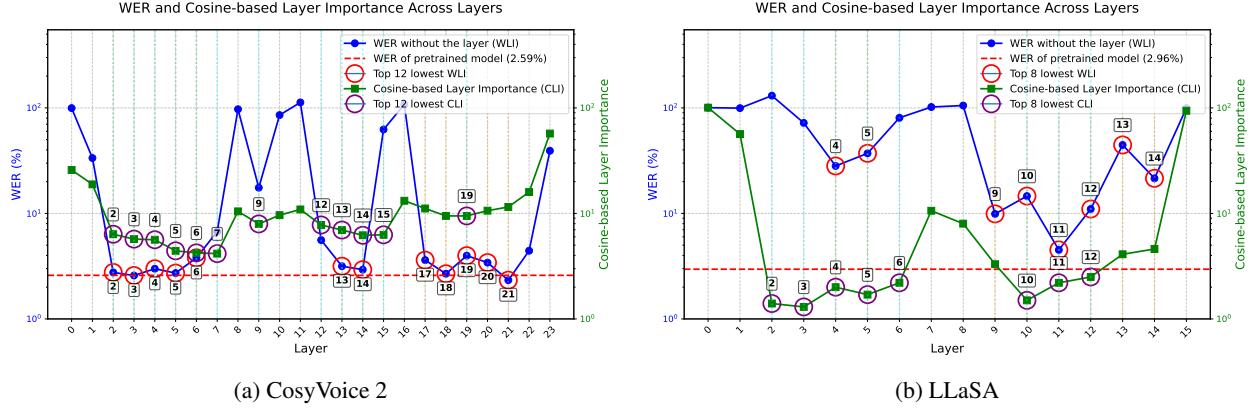


Fig. 2: WLI and cosine-based layer importance of (a) CosyVoice 2 and (b) LLaSA. High WLI indicates that WER increases significantly when the layer is removed, and high cosine-based importance indicates the input and output latents of the layer are dissimilar. We found that, based on WLI, the layers in the beginning, middle, and the end contributes critically to the performance. Our method prunes the model by removing layers with least contribution to the performance.

through a Word Error Rate (WER) based layer importance index and recovering performance via multi-level distillation, SPADE achieves substantial efficiency gains while preserving perceptual quality. Across zero-shot benchmarks, SPADE halves Transformer depth, reduces overall parameters by up to 40%, lowers VRAM consumption by up to 20%, and accelerates inference by as much as $1.7\times$, all while retaining near-parity in perceptual metrics. Moreover, the knowledge distillation process shows remarkably high data-efficiency: The recovery of performance requires less than 5% of the pretraining data of the original checkpoints. These results highlight the framework of pruning and distillation as a practical pathway toward compact, high-fidelity, real-time speech generation.

2. PROPOSED METHOD

Our framework first explores the importance of each layer in LLM backbone and prune non-essential layers to compress the model. Subsequently, an efficient knowledge distillation is applied to effectively restore the performance of the pruned model.

2.1. Model compression through pruning

SPADE is motivated by using the nature of residual connections: $x^l = x^{l-1} + f_l(x^{l-1})$, where x^l and f^l denote the hidden state and transformation in the layer l , respectively. In LLM-TTS, each transformer layer contributes by refining latent representations through residual connections [22]. Although the architecture shows strong performance in diverse applications, recent studies in text-based LLM suggest some layers provide

only weak refinements and can be removed with little effect [13, 23]. An established criterion to identify them is to compute the cosine distance between inputs and outputs of a layer [24]. However, Fig. 2 shows that cosine-based layer importance (CLI) does not align with the performance contribution of layers in TTS, as indicated by the different patterns in WER change.

To address this, we leverage WER, as the primary interest is the semantic consistency of generation, and propose WER-based layer importance (WLI):

$$\text{WLI}_i = \mathbb{E}_{\mathcal{D}} \left[\text{WER}(\text{model}(\mathbf{x}_2; \boldsymbol{\theta}_{\setminus i}, \mathbf{x}_1, \mathbf{y}_1), \mathbf{y}_2) \right], \quad (1)$$

where $\theta_{\setminus i}$ denotes the model parameters without the i -th layer, \mathcal{D} is a subset of the evaluation set, and (x_1, y_1) and (x_2, y_2) denote reference and query text-audio pairs, respectively. Specifically, a layer is considered important only if its absence causes significant degradation in WER. Unlike cosine-based layer importance, which only estimates the difference between inputs and outputs of within a layer, WLI directly measures the contribution of each layer to the final performance. As shown in Fig 2 evaluated WLI with Whisper [25], and found that many of the layers have negligible impact on performance, indicating a significant redundancy and aligning with the findings in text-domain LLMs [13]. Moreover, we find that the earliest, central, and final layers consistently emerge as important across different LLM-TTS models. Based on the analysis, we prune transformer layers with low WLI values from the LLM backbones.

2.2. Recovering original performance

While the proposed framework effectively reduces the parameters without additional modules, such as parameter-

efficient fine-tuning [26, 27], the pruned model naturally confronts disconnected flow of latent information. To address this, we leverage the original un-pruned model as the teacher and perform a knowledge distillation training that simply *heals* the pruned model to minimize the loss of performance without any additional parameters. To maximize the restoration, we employ a composite loss that benefits from both supervised learning and teacher-guided knowledge distillation:

$$\mathcal{L} = \alpha * \mathcal{L}_{CE} + \frac{1 - \alpha}{4} * (\mathcal{L}_{logit} + \mathcal{L}_l + \mathcal{L}_a + \mathcal{L}_e). \quad (2)$$

Here, the Cross-Entropy loss \mathcal{L}_{CE} is responsible for the **supervised** component that directly guides the output distribution. The **distillation** component comprises 4 elements: Embedding reconstruction loss \mathcal{L}_e , alignment losses on logit \mathcal{L}_{logit} , latent \mathcal{L}_l , and attention \mathcal{L}_a , following [24]. To provide more stability, we implement the \mathcal{L}_{logit} with mixed distribution by leveraging Skew KL Divergence [16]. For \mathcal{L}_e , \mathcal{L}_l , and \mathcal{L}_a , we calculate Mean Squared Error (MSE) of embedding outputs, intermediate latents, and attention matrices between the teacher and student, respectively. To maximize the distillation of teacher’s knowledge, we propose to dynamically select layers to apply loss for \mathcal{L}_l and \mathcal{L}_a . Specifically, as shown in Fig 1, the target values (latent, attention map) for the student layer (l_n) are derived from the last layer before the next retained layer (l_{m+2}) from the teacher model. While simple, this approach allows the pruned student model to retain not only its original capability but also those of the removed layers, culminating in a smaller yet more compact model. Finally, we combine the supervised and distillation components by adjustable weight *alpha* to balance the influence of each, where the value is empirically set to 0.25 to deliberately provide stronger supervised guidance.

3. EXPERIMENTAL SETUP

To evaluate whether pruning and distillation can truly enable compact yet high-fidelity LLM-TTS systems, we benchmark SPADE against two representative baselines: CosyVoice 2² [28] and LLaSA-1B³ [6]. Both models are chosen for their strong zero-shot capabilities and public availability of checkpoints.

Since a key motivation of SPADE is to minimize re-training cost, we fine-tune each pruned variant on only a fraction of the data: 25% of LibriHeavy [29] (EN) for LLaSA and LibriTTS [30] (EN) for CosyVoice 2. This corresponds to less than 5% of the pretraining corpus size, testing whether our framework can recover quality under data-constrained conditions.

²<https://github.com/FunAudioLLM/CosyVoice>

³https://github.com/zhenye234/LLaSA_training

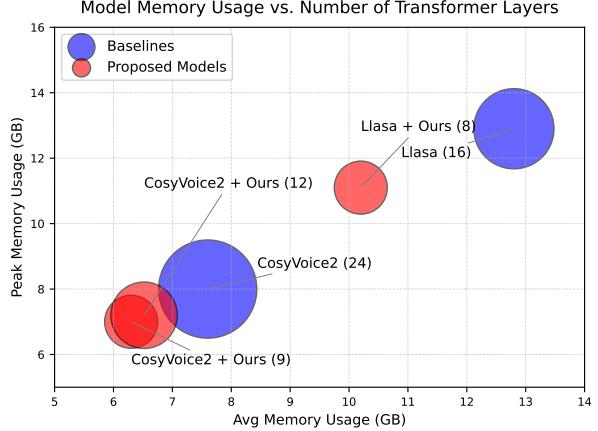


Fig. 3: SPADE effectively reduces half of the Transformer layers, reducing VRAM usage by 14% for CosyVoice 2 and 20% for LLaSA.

We assess the performance on *LibriTTS test-clean* and the *Seed-TTS eval set* [31], both of which are widely used for zero-shot evaluation. All experiments are conducted on 4×NVIDIA A6000 GPUs, with official training scripts left unchanged aside from data size and pruning. CosyVoice 2 is fine-tuned with dynamic batches up to 20,000 tokens, whereas LLaSA uses a batch size of 4. Fine-tuning with SPADE runs for 7 epochs to CosyVoice 2 and for 1 epoch when applying it to LLaSA.

To evaluate both computational efficiency and perceptual quality, we consider a range of complementary metrics. Efficiency is assessed in terms of model depth, parameter count, and real-time factor (RTF), while intelligibility is measured using word error rate (WER). Perceptual aspects are captured objectively through speaker similarity (SS) and UTMOS using VERSA toolkit [32], and subjectively through the naturalness mean opinion score (NMOS) with 20 listening and 50 random samples from both evaluation sets per model. Together, these metrics reveal whether compact models preserve the qualities essential for real-world TTS deployment.

4. RESULTS AND ANALYSIS

4.1. Effect on performance after pruning

Table 1 presents the main results of applying SPADE to CosyVoice 2 and LLaSA with different configurations. For CosyVoice, pruning to 12 layers halves the depth, reduces parameters by 39.7% and accelerates inference by 42.6%. Moreover, effective VRAM usage is reduced by 14%, as depicted in Fig. 3. Notably, these gains come with no significant degradation in both quantitative and qualitative metrics, showing slight increase of 0.68 in WER for challenging Seed-TTS dataset and 0.11 reduction in NMOS. A more aggressive variant with only 9

Model	Layers	Params	RTF	NMOS	Seed-TTS <i>test-en</i>			LibriTTS <i>test-clean</i>		
					WER	SS	UTMOS	WER	SS	UTMOS
Human Record	-	-	-	3.96 ± 0.14	1.47	1.00	3.52	1.85	1.00	4.14
Vocoder Resyn. of GT	-	-	-	3.87 ± 0.15	1.53	1.00	3.53	1.65	1.00	4.01
CosyVoice 2 ⁴ [28]	24	0.63	0.61	3.71 ± 0.13	2.03	0.66	4.15	1.43	0.81	4.41
CosyVoice 2 + Ours	12	0.38	0.35	3.58 ± 0.14	2.71	0.66	4.16	1.59	0.82	4.41
CosyVoice 2 + Ours	9	0.32	0.33	3.55 ± 0.14	3.09	0.66	4.15	1.94	0.81	4.40
Codec Resyn. of GT	-	-	-	3.59 ± 0.15	2.49	1.00	3.69	2.52	1.00	4.01
LLaSA ⁴ [6]	16	1.7	0.82	3.37 ± 0.15	3.54	0.46	4.13	1.54	0.47	4.41
LLaSA + Ours	8	1.3	0.58	3.11 ± 0.14	4.20	0.41	4.06	1.88	0.43	4.40

(a) Absolute zero-shot evaluation results on Seed-TTS Eval set and LibriTTS test-clean. Higher values indicate better performance for NMOS, SS, and UTMOS, while lower values indicate better performance for RTF and WER. Missing entries correspond to trivial cases, e.g., human records or codec/vocoder resynthesis of groundtruth (GT).

Model Pair	Layers	Params	RTF	Data	WER	SS	UTMOS
CosyVoice2 → CosyVoice2 + Ours (12)	↓ 50.0%	↓ 39.7%	↓ 42.6%	2(0.3)% ↓	+0.43 ↑	+0.005 ↑	0.00 ↔
CosyVoice2 → CosyVoice2 + Ours (9)	↓ 62.5%	↓ 49.2%	↓ 45.9%	2(0.3)% ↓	+0.79 ↑	+0.000 ↔	0.00 ↔
LLaSA → LLaSA + Ours	↓ 50.0%	↓ 23.5%	↓ 29.3%	≤ 12.5(5)% ↓	+0.50 ↑	-0.045 ↓	-0.04 ↓

(b) Relative performance of SPADE models compared to their uncompressed versions. For data usage, the exact amount of English data in the internal LLaSA pretraining set is unknown; we therefore report an upper bound (\leq).

Table 1: Zero-shot evaluation on Seed-TTS (*test-en*) and LibriTTS (*test-clean*). Table 1a shows efficiency and quality, while Table 1b reports relative gaps of pruned variants.

layers further reduces parameters by 49.2%, improves RTF by 45.9%, and lowers VRAM usage by 17%. While this extreme setting renders additional increase in WER, perceptual metrics such as NMOS, SS, and UTMOS remain stable, suggesting that SPADE enables flexible trade-offs between efficiency and intelligibility. To confirm that our framework generalizes beyond a single backbone, we conduct experiment on LLaSA, a larger variant based on speech codec. Here, pruning removes half the layers, decreases parameters by 23.5%, improves RTF by 29.3% (1.41 \times speed-up), and reduces VRAM by 20%. The result suggests that, while most of the metrics lie in acceptable range, the performance degradation is relatively larger compared to CosyVoice 2. Based on the analysis in Fig 2⁴, we expect it attributes to overall high WLI values across all layers, indicating each layer contributes similarly to the performance.

4.2. Ablation study

We evaluate the effectiveness of the proposed methodology with systematic ablation study. Table 2 presents experiments on CosyVoice 2 with LibriTTS test-clean. First, the proposed WLI-based pruning is replaced with existing cosine-based pruning. The result shows a notable increase in WER and CER, demonstrating the pruning based on WLI, a metric directly related to intelligi-

Expr.	WER ↓	CER ↓	SS ↑	UTMOS ↑
CosyVoice 2	1.43	0.46	0.81	4.41
CosyVoice 2 + Ours	1.59	0.54	0.82	4.41
Cosine-based pruning	1.74	0.61	0.81	4.40
Distill from original layer	1.65	0.58	0.81	4.40

Table 2: Ablation experiment on LibriTTS test-clean. Both cosine-based pruning and the alternative latent knowledge distillation scheme degrade the overall performance, where cosine-based pruning shows more significant increase in WER and CER.

gibility, successfully prevents performance degradation. Moreover, dynamic distillation loss is removed and student layer is distilled only with the information from the corresponding teacher layer. The overall decrease in performance suggests that adaptively choosing the target in distillation training is simple yet plays a significant role.

5. CONCLUSION

We presented SPADE, a pruning-based framework for compressing LLM-TTS models, and showed that SPADE achieves substantial efficiency gains while preserving intelligibility and naturalness. We investigate the importance of each layer using the proposed WLI and find that many layers contribute little to audio synthesis. Experimental results verify that, by applying SPADE, such layers can be removed without harming perceived quality.

⁴Metrics are reported based on official checkpoints

6. REFERENCES

- [1] Zhihao Du, Qian Chen, Shiliang Zhang, et al., “CosyVoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens,” *arXiv:2407.05407*, 2024.
- [2] Zhihao Du, Changfeng Gao, Yuxuan Wang, et al., “CosyVoice 3: Towards in-the-wild speech generation via scaling-up and post-training,” *arXiv:2505.17589*, 2025.
- [3] Chengyi Wang, Sanyuan Chen, Yu Wu, et al., “Neural codec language models are zero-shot text to speech synthesizers,” *arXiv:2301.02111*, 2023.
- [4] Jaehyeon Kim, Keon Lee, Seungjun Chung, and Jaewoong Cho, “CLam-TTS: Improving neural codec language model for zero-shot text-to-speech,” in *Proc. ICLR*, 2024.
- [5] Detai Xin, Xu Tan, Kai Shen, Zeqian Ju, et al., “RALL-E: Robust codec language modeling with chain-of-thought prompting for text-to-speech synthesis,” *arXiv:2404.03204*, 2024.
- [6] Zhen Ye, Xinfu Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, et al., “LLaSA: Scaling train-time and inference-time compute for llama-based speech synthesis,” *arXiv:2502.04128*, 2025.
- [7] Zalán Borsos, Raphaël Marinier, Damien Vincent, et al., “AudioLM: a language modeling approach to audio generation,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 31, pp. 2523–2533, 2023.
- [8] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al., “LLaMA: Open and efficient foundation language models,” *arXiv:2302.13971*, 2023.
- [9] Shuai Bai, Keqin Chen, Xuejing Liu, et al., “Qwen2. 5-vl technical report,” *arXiv:2502.13923*, 2025.
- [10] Dong Zhang, Shimin Li, Xin Zhang, et al., “SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities,” *arXiv:2305.11000*, 2023.
- [11] Dongkuan Xu, Ian EH Yen, Jinxi Zhao, and Zhibin Xiao, “Rethinking network pruning—under the pre-train and fine-tune paradigm,” *arXiv:2104.08682*, 2021.
- [12] Zejian Liu, Fanrong Li, Gang Li, and Jian Cheng, “EBERT: Efficient bert inference with dynamic structured pruning,” in *Findings of the ACL: ACL-IJCNLP*, 2021, pp. 4814–4823.
- [13] Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts, “The unreasonable ineffectiveness of the deeper layers,” *arXiv:2403.17887*, 2024.
- [14] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu, “Patient knowledge distillation for bert model compression,” *arXiv:1908.09355*, 2019.
- [15] Haojie Pan, Chengyu Wang, Minghui Qiu, et al., “Meta-KD: A meta knowledge distillation framework for language model compression across domains,” *arXiv:2012.01266*, 2020.
- [16] Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun, “DistiLLM: Towards streamlined distillation for large language models,” *arXiv:2402.03898*, 2024.
- [17] Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat, “Q8BERT: Quantized 8bit bert,” in *EMC2-NIPS*, 2019.
- [18] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, et al., “ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers,” *NeurIPS*, 2022.
- [19] Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi, “SliM-LLM: Salience-driven mixed-precision quantization for large language models,” *arXiv:2405.14917*, 2024.
- [20] Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin, “DeeBERT: Dynamic early exiting for accelerating bert inference,” *arXiv:2004.12993*, 2020.
- [21] Xinyin Ma, Gongfan Fang, and Xinchao Wang, “LLM-Pruner: On the structural pruning of large language models,” *NeurIPS*, 2023.
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proc. CVPR*, 2016.
- [23] Andreas Veit, Michael J Wilber, and Serge Belongie, “Residual networks behave like ensembles of relatively shallow networks,” *NeurIPS*, 2016.
- [24] Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, et al., “Compact language models via pruning and knowledge distillation,” *NeurIPS*, 2024.
- [25] Alec Radford, Jong Wook Kim, Tao Xu, et al., “Robust speech recognition via large-scale weak supervision,” in *Proc. ICML*, 2023.
- [26] Edward J Hu, Yelong Shen, Phillip Wallis, et al., “LoRA: Low-rank adaptation of large language models.,” *Proc. ICLR*, 2022.
- [27] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer, “QLoRA: Efficient finetuning of quantized llms,” *NeurIPS*, 2023.
- [28] Zhihao Du, Yuxuan Wang, Qian Chen, et al., “CosyVoice 2: Scalable streaming speech synthesis with large language models,” *arXiv:2412.10117*, 2024.
- [29] Wei Kang, Xiaoyu Yang, Zengwei Yao, et al., “Libri-heavy: A 50,000 hours asr corpus with punctuation casing and context,” in *Proc. ICASSP*, 2024.
- [30] Heiga Zen, Viet Dang, Rob Clark, et al., “LibriTTS: A corpus derived from librispeech for text-to-speech,” *arXiv:1904.02882*, 2019.
- [31] Philip Anastassiou, Jiawei Chen, Jitong Chen, et al., “Seed-TTS: A family of high-quality versatile speech generation models,” *arXiv:2406.02430*, 2024.
- [32] Jiatong Shi, Jinchuan Tian, Yihan Wu, et al., “ESPnet-Codec: Comprehensive training and evaluation of neural codecs for audio, music, and speech,” in *IEEE Spoken Language Technology workshop*, 2024.