VocalNet: Speech LLMs with Multi-Token Prediction for Faster and High-Quality Generation

Anonymous ACL submission

Abstract

Speech large language models (LLMs) have emerged as a prominent research focus in speech processing. In this work, we introduce **VocalNet**, a series of high-performance speech LLMs featuring a scalable and model-agnostic training framework as well as a novel multi-token prediction (MTP) paradigm for speech generation. We first propose an efficient two-stage training framework that enables LLMs to acquire real-time speech interaction capabilities. Through extensive experiments on various training configurations, we ensure both simplicity and effectiveness in the training strategy. Furthermore, inspired by advances in language modeling, we introduce MTP into the domain of speech LLMs—an alternative to traditional next-token prediction (NTP)—which enables the model to predict multiple future tokens at each step. Through systematic analysis and improved implementation, we show that MTP not only accelerates inference speed but also significantly enhances speech quality. Experimental results demonstrate that VocalNet achieves performance comparable to state-of-the-art Omni LLMs while outperforming existing open-source speech LLMs, despite using limited training data.

1 Introduction

The evolution of speech interaction systems has progressed from traditional cascade architectures to modern end-to-end approaches. While conventional systems employ separate modules for automatic speech recognition (ASR), large language model (LLM), and text-to-speech (TTS) (Shen et al., 2023; Huang et al., 2024; An et al., 2024), these pipeline systems suffer from latency accumulation and information degradation. Recent breakthroughs like GPT-4o (OpenAI, 2024) have demonstrated the superior potential of end-to-end speech LLMs that process audio inputs and outputs directly within a unified framework, enabling more natural and responsive voice interactions.

Current speech LLMs can be broadly categorized into two paradigms (Chen et al., 2025). Native multimodal models like Mini-Omni (Xie and Wu, 2024) and Moshi (Défossez et al., 2024) employ a decoder-only Transformer for joint text and speech processing, but require massive pretraining data and face catastrophic forgetting issues. In contrast, aligned multimodal models such as LLaMA-Omni (Fang et al., 2024) and Freeze-Omni (Wang et al., 2024) preserve LLM capabilities through separate speech encoders and decoders while requiring less training data. Despite notable advances in aligned multimodal speech LLMs, two critical challenges severely limit their widespread adoption and real-world deployment.

First, the design of training frameworks for aligned models remains underdeveloped and excessively complex. Systems like Freeze-Omni and MinMo (Chen et al., 2025) employ complex multi-stage training procedures whose empirical benefits are unclear, introducing computational overhead and reproducibility challenges. This complexity not only slows down research progress but also raises barriers for practical scalability and industrial application. Second, the prevailing autoregressive next-token prediction (NTP) paradigm (Zeng et al., 2024; Xu et al., 2025) inherently constrains both the efficiency and quality of speech generation. Its sequential token-by-token generation leads to inference latency, incompatible with real-time or large-scale scenarios. More critically, NTP's token-level granularity is poorly aligned with the hierarchical structure of speech, since meaningful acoustic units like phonemes or syllables typically span multiple tokens. This structural mismatch undermines model training efficiency and directly impacts the naturalness and intelligibility of generated speech. Although non-autoregressive alternatives using CTC loss (Fang et al., 2024; Luo et al., 2025) offer accelerated generation, they typically do so at a substantial cost to output fidelity, failing to fully

resolve the limitations posed by NTP. Addressing these fundamental challenges is essential to unlocking the full potential of speech LLMs in practical, high-impact applications.

Therefore, this paper introduces VocalNet, a breakthrough in speech LLMs that simultaneously addresses training efficiency and improve speech generation through two key innovations. First, we propose a scalable, LLM-agnostic two-stage training framework designed to efficiently equip LLMs with real-time speech interaction capabilities using limited data. We experimentally investigated various configurations for this framework, including the necessity of pretraining stages and comparing the performance of single-stage versus two-stage training. Based on these experimental results, we established a streamlined two-stage training framework that maintains effectiveness while preserving simplicity. Furthermore, aiming to simultaneously enhance generation speed and speech quality, and inspired by recent advances in language modeling (Qi et al., 2020; Gloeckle et al., 2024; Cai et al., 2024), we explore the potential of multi-token prediction (MTP) in the context of speech LLMs. Through careful analysis of the impact of MTP on speech generation, we identify limitations in existing approaches and propose a more effective implementation specifically for speech LLMs. Our findings demonstrate that, even with limited training data, this MTP method not only accelerates generation speed but also significantly improving speech quality (~50% WER reduction) compared to conventional NTP. Leveraging the proposed training framework and MTP method, we successfully trained VocalNet-1B and VocalNet-8B. Experimental results demonstrate that VocalNet-1B significantly outperforms existing speech LLMs of comparable parameter size. VocalNet-8B achieves performance on par with advanced Omni LLMs like MiniCPM-o (Yao et al., 2024) and Qwen2.5-Omni (Xu et al., 2025), despite utilizing considerably less training data. Moreover, VocalNet-8B markedly surpasses previous open-source speech LLMs such as Freeze-Omni (Wang et al., 2024). Our key contributions are as follows:

- **Effective Training Framework for Speech LLMs.** We propose a scalable and model-agnostic training framework that efficiently integrates speech understanding and generation capabilities into LLMs through the incorporation of a speech encoder and decoder.

- **Multi-Token Prediction (MTP): A Paradigm Shift for Enhanced Speech Generation.** We identify inefficiencies in standard next-token prediction (NTP) for speech generation and propose multi-token prediction (MTP) as a novel paradigm for autoregressive speech modeling. Our optimized MTP implementation not only accelerates inference but also improves output quality, offering new insights into efficient speech generation.

- **High-Performance Speech LLMs: VocalNet Models.** Leveraging the proposed training framework and MTP method, we develop high-performance speech LLMs—VocalNet. Experimental results show their strong performance in voice interaction tasks even with limited training data, demonstrating the effectiveness and efficiency of our approach.

2 VocalNet

2.1 Model Architecture

The model architecture of VocalNet is illustrated in Figure 1. Align with prior work, VocalNet consists of a speech encoder to convert waves into speech representations, a LLM backbone and a speech decoder for speech tokens generation. A downsample adaptor is added after the speech encoder to achieve a lower frame rate, and a speech projector to bridge the dimension gap between the LLM hidden state and decoder inputs. The generated speech token is sent to the speech vocoder, in which the corresponding speech response is constructed. This architecture effectively preserves the capabilities inherent in the LLM, thus significantly reducing the data requirement for training compared with native multimodal models. In the following statement, $\boldsymbol{x}^s$ refers to the raw speech query, $\boldsymbol{y}^t$ represents the generated text response and $\boldsymbol{y}^s$ stands for the speech response.

Speech Query Encoding The speech encoder E processes the input speech query $\boldsymbol{x}^s$ to produce a high-level representation $\boldsymbol{z}$ with length l: $\boldsymbol{z} = E(\boldsymbol{x}^s) = (z_0, z_1, ..., z_l)$, which encapsulates rich semantic information. After that, the downsample adaptor transforms the speech feature $\boldsymbol{z}$ into semantic-condensed embedding with a lower frame rate. Through a concatenation-based projection module, it reduces the sequence length by a

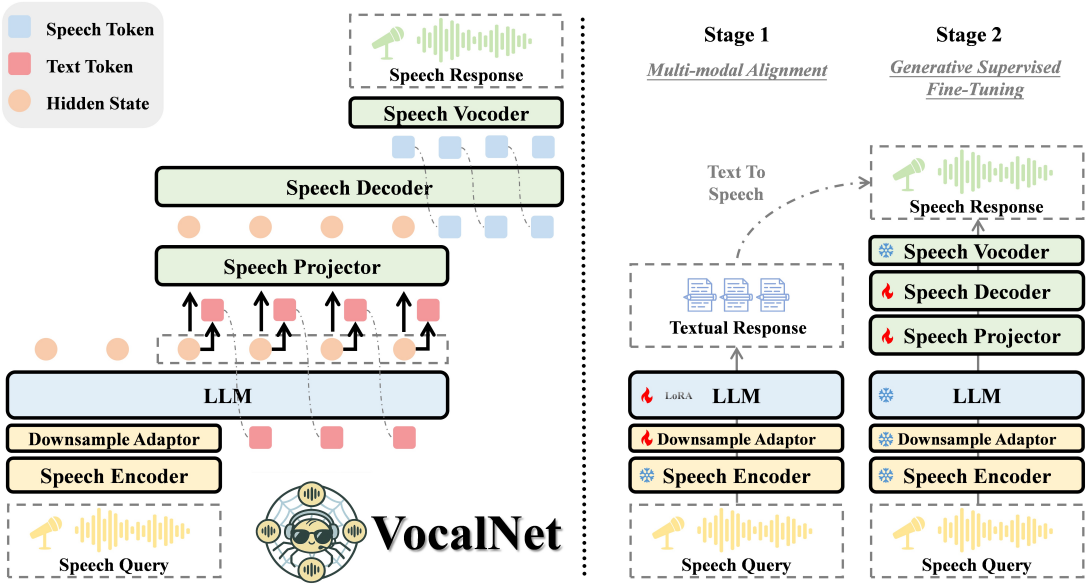

Figure 1: On the **left**: The architecture of the VocalNet model. On the **right**: A depiction of VocalNet's dual-stage training strategy.

factor of k, yielding $\boldsymbol{z}'$, and applies linear transformations with ReLU to generate $\boldsymbol{z}_o$, which will be fed into the LLM backbone, as expressed in

$$\begin{aligned} \boldsymbol{z}'_i &= \mathrm{Concat}(z_{ir}, z_{ir+1}, \ldots, z_{(i+1)r-1}) \\ \boldsymbol{z}_o &= W_2(\mathrm{ReLU}(W_1 \boldsymbol{z}' + b_1)) + \boldsymbol{b}_2 \end{aligned} \tag{1}$$

where W_1 and W_2 are weight matrices, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are bias vectors.

LLM The LLM functions as the core module, processing the compressed representation $\boldsymbol{z}_o$ to extract linguistic and contextual information, yielding hidden states $\boldsymbol{h}_{LLM}$. These states enable the generation of the corresponding textual response $\boldsymbol{y}^t$ and are essential in speech generation.

Speech Response Generation The speech decoder need to model both the LLM hidden states $\boldsymbol{h}_{LLM}$ and the speech embedding simultaneously, but the spaces represented by these two are typically different (Wang et al., 2024). To address this space gap, we use a speech projector that transforms $\boldsymbol{h}_{LLM}$ into $\boldsymbol{v}_{LLM}$. The speech decoder then utilizes these vectors to autoregressively generate a sequence of discrete speech tokens $\boldsymbol{s}$. Finally, a pre-trained speech vocoder, incorporating a chunk-aware flow matching model derived from CosyVoice2 (Du et al., 2024) along with HiFiGAN (Kong et al., 2020), constructs the mel-spectrogram from the speech tokens $\boldsymbol{s}$ and then synthesizes the speech waveform response $\boldsymbol{y}^s$. VocalNet also supports streaming speech generation; the detailed implementation is described in Appendix B.

2.2 Training Strategy

We adopt a dual-stage training strategy (see the right part of Figure 1), consisting of Multi-Modal Alignment and Generative Supervised Fine-Tuning, following the categorization in (Ji et al., 2024). In the first stage, VocalNet is trained on speech-to-text tasks ($\boldsymbol{x}^s \rightarrow \boldsymbol{y}^t$). The speech encoder is kept frozen to preserve its speech representation capability, while the downsample adaptor is updated to align speech and text features. The LLM backbone is fine-tuned using LoRA to enhance multi-modal understanding, without compromising its original knowledge and reasoning abilities. A cross-entropy loss on text tokens is used to guide learning. In the second stage, VocalNet is trained on speech-to-speech tasks ($\boldsymbol{x}^s \rightarrow \boldsymbol{y}^s$). Most model components are frozen, and only the speech projector and speech decoder are updated to generate high-quality speech tokens s that match the ground-truth response y^s. A cross-entropy loss on speech tokens is applied for training.

Our staged training approach decomposes the task into two manageable steps, allowing for a more stable and controlled training process. While our framework could support training both speech understanding and generation within a single stage, our experiments reveal that single-stage training negatively impacts the performance of speech

LLMs in spoken QA, without offering clear advantages over the two-stage approach in terms of speech generation quality. Moreover, we find that pretraining with separate ASR and TTS tasks does not yield significant performance improvements, yet introduces additional computational costs. As a result, our framework excludes any dedicated pretraining stages. We provide a detailed discussion of various training framework configurations and their experimental results in Section 5.2.

3 Multi-Token Prediction for Speech Generation

3.1 Motivation

Current speech LLMs predominantly adopt next-token prediction (NTP) within an autoregressive (AR) framework (Fang et al., 2024; Wang et al., 2024), generating speech token-by-token. While this approach has achieved notable success, it faces several fundamental limitations due to the unique characteristics of speech signals, suggesting that NTP may not be the most efficient or optimal strategy for speech generation. First, speech tokens exhibit a much higher temporal resolution (~25Hz (Du et al., 2024)) compared to text tokens (~3Hz (Li et al., 2025a; Défossez et al., 2024)), resulting in significantly longer sequences. The sequential nature of NTP—predicting one token at a time—inherently limits generation speed and introduces latency, which poses a major challenge for real-time voice interaction systems. Additionally, human speech exhibits a hierarchical acoustic-semantic structure encompassing phonemes, syllables, and prosody, operating over timescales longer than individual speech tokens (e.g., 40ms segments in CosyVoice2 (Du et al., 2024)). Unlike text tokens, which correspond to discrete semantic units, speech tokens often lack independent meaning and must be jointly modeled to capture linguistically coherent patterns. The myopic focus of NTP on predicting single tokens struggles to learn such inter-token dependencies—particularly under limited training data—leading to suboptimal modeling of the rich temporal dynamics inherent in speech.

Inspired by recent advances in LLMs (Gloeckle et al., 2024; Li et al., 2024; Cai et al., 2024), we introduce multi-token prediction (MTP) for speech generation. MTP addresses the limitations discussed above by modeling the joint distribution of multiple tokens, thereby compressing sequence generation into fewer steps. This leads to two key benefits: significantly reduced inference steps for faster generation, and improved modeling of long-range dependencies and speech's hierarchical structure. As a result, MTP enhances both efficiency and output quality, offering a promising direction for speech generation. In the following section, we first analyze the limitations of previous related MTP approaches and then present a novel implementation that is both simple in design and highly effective for speech generation.

3.2 Implementation of MTP

Group Modeling Method To accelerate speech token generation, prior work has employed the Group Modeling method (Chen et al., 2024; Zhang et al., 2024b) to enable multi-token prediction, as illustrated in Figure 2(a). This approach divides the speech token sequence into fixed-size groups, merges tokens within each group into a single embedding, and processes these embeddings through the backbone. A decomposition layer then reconstructs the original tokens from each group embedding. SLAM-Omni (Chen et al., 2024) uses a linear layer for decomposition, while IntrinsicVoice (Zhang et al., 2024b) employs a non-autoregressive Transformer with learnable queries. However, these methods often degrade speech quality due to information loss and disruption of intra-group temporal dependencies. Moreover, the fixed group size restricts dynamic control over generation speed during inference.

MTP Implementation in LLMs Inspired by the implementation of MTP in Gloeckle et al. (2024) and DeepSeek-V3 (Liu et al., 2024), we designed two speech decoder architectures to achieve multi-token prediction: MTP-Parallel-Linear and MTP-DeepSeek. As shown in Figure 2(b), MTP-Parallel-Linear predicts n future tokens in parallel using independent linear heads. While efficient and commonly used in LLMs, this approach fails to explicitly model temporal dependencies among speech tokens—crucial for capturing the continuous and sequential nature of speech. This limitation often results in reduced output coherence, especially as the number of prediction heads increases.

In contrast, MTP-DeepSeek generates tokens sequentially, preserving causal dependencies at each depth (Figure 2(c)). However, during training, this method inputs the ground truth $x_{\leq i+k}$ to the k-th MTP module to predict x_{i+k+1} and computes the loss of a teacher-forced next-step prediction. Con-

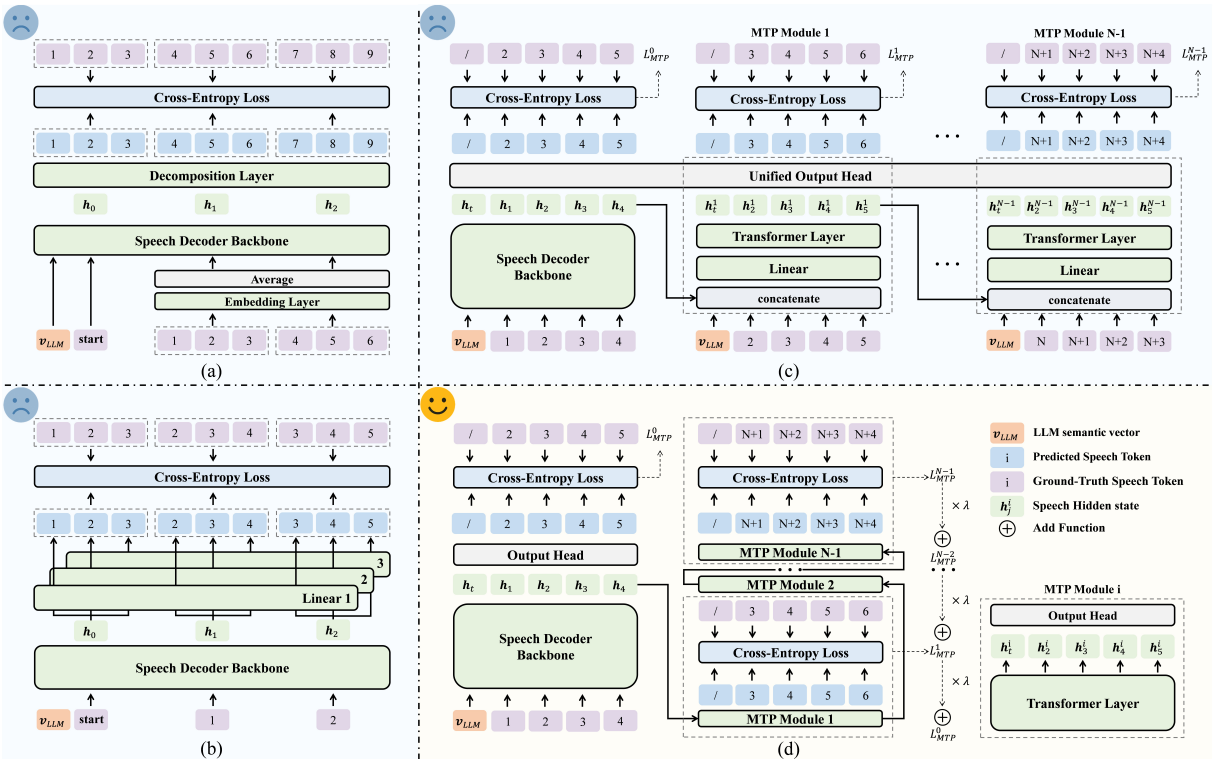

Figure 2: Illustration of various accelerate implementations. (a): Group Modeling; (b): MTP-Parallel-Linear; (c): MTP-DeepSeek; (d): Our MTP implementation.

sequently, this implementation actually optimizes the loss function $-\sum_{\boldsymbol{x}}\sum_k \log q(\boldsymbol{x}_{t+k+1}|\boldsymbol{x}_{\leq t+k})$, which is essentially the same as the NTP loss. As a result, despite enabling multi-token prediction, it does not effectively help capture local speech patterns or alleviate error accumulation. We further analyze this limitation in Section 5.3 and Appendix C.

Our MTP Implementation Based on our analysis of existing MTP methods, we propose a simple yet more effective MTP implementation for speech LLMs. Given the sequential nature of speech and its reliance on temporal coherence, our method—illustrated in Figure 2(d)—employs $N-1$ sequential Transformer layers as MTP modules to predict N future speech tokens in a single step, while preserving causal dependencies between them. Unlike MTP-DeepSeek, our approach uses previously computed hidden states instead of ground-truth tokens as input. Let $\boldsymbol{h}^0_{1:(L_t+t)}$ denote the initial hidden state generated by the speech decoder backbone, conditioned on the v_{LLM} and t history tokens. This state is sequentially passed through $N-1$ MTP modules:

$$\boldsymbol{h}^k_{1:(L_t+t)} = \mathrm{MTP}_k(\boldsymbol{h}^{k-1}_{1:(L_t+t)}) \tag{2}$$

where $\boldsymbol{h}^k_{1:(L_t+t)}$ represents the hidden state output of the k-th MTP module. This layer-wise propagation preserves the causal dependencies of the speech sequence. The resulting N hidden states at index L_t+t, $\boldsymbol{h}^0_{L_t+t}, \boldsymbol{h}^1_{L_t+t}, \ldots, \boldsymbol{h}^{N-1}_{L_t+t}$, are then fed into N independent output heads to produce token predictions:

$$\begin{aligned} p^k_{t+k+1} &= \mathrm{OutHead}_k(\boldsymbol{h}^k_{L_t+t}) \\ &= \mathrm{Linear}_k(\mathrm{RMSNorm}(\boldsymbol{h}^k_{L_t+t})) \end{aligned} \tag{3}$$

where $k \in \{0, 1, \ldots, N-1\}$, and p^k_{t+k+1} denotes the predicted probability distribution for the $(t+k+1)$-th token.

To train this architecture, we minimize the prediction error across all depths of the MTP modules. Specifically, the loss is defined as a weighted average of cross-entropy losses from each output head:

$$\mathcal{L}_{mtp} = \sum_{k=0}^{N-1} \lambda^k \, \mathrm{CE}(p^k_{k+1:L_s}, s_{k+1:L_s}) \tag{4}$$

where L_s is the total speech sequence length, $\mathrm{CE}(\cdot)$ denotes the cross-entropy loss, and $s_{k+1:L_s}$ denotes the ground-truth tokens from index $k+1$ to L_s. Here, the decay factor $\lambda \in (0,1)$ controls the importance of predictions at different

depths of the MTP modules. Specifically, it assigns higher weights λ^k to the losses from earlier layers (smaller k), as these layers typically produce more reliable and immediate predictions. Conversely, losses from deeper layers (larger k), which tend to have higher uncertainty, receive progressively lower weights λ^k. This prioritizes short-term accuracy while still leveraging long-range context modeling.

4 Experiments Setup

4.1 Datasets

The training data for VocalNet combines VoiceAssistant-400K from Mini-Omni and UltraChat from SLAM-Omni (Xie and Wu, 2024; Chen et al., 2024). VoiceAssistant-400K contains approximately 470K samples generated by GPT-4o; after removing instances with overly long responses, we retain 430K query-response pairs. For UltraChat, we split multi-round dialogues into single rounds due to missing initial turns and weak contextual links, resulting in approximately 300K samples. Speech responses for both datasets are synthesized using CosyVoice2-0.5B (Du et al., 2024). In total, VocalNet is trained on 730K examples, corresponding to approximately 6,000 hours of speech—substantially less than other advanced models such as Baichuan-Omni-1.5 (887K hours of multi-modal pretraining) and Minmo (around 1.4M hours).

4.2 Model Configuration

We propose VocalNet-1B and VocalNet-8B built upon LLaMA-3.2-1B-Instruct [1] and LLaMA-3.1-8B-Instruct [2] respectively. Both models employ Whisper-large-v3 (Radford et al., 2023) as the speech encoder, and use the flow-matching model and HiFi-GAN vocoder from CosyVoice2 for speech synthesis. A two-layer linear downsample adaptor reduces feature dimension with a factor of 5. The speech projector consists of two LLaMA decoder layers, while the speech decoder contains four. Each MTP module is implemented with a single LLaMA decoder layer followed by a linear output head.

[1] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct

[2] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

4.3 Training and Evaluation Details

VocalNet is trained in two stages: the first focuses on the downsample adaptor and LLM, while the second trains the speech projector and decoder. Both stages use a learning rate of 2×10^{-4} with cosine annealing and a warmup ratio of 0.03. All experiments are conducted on A100 GPUs.

To evaluate the capabilities of voice interaction, we utilize the English subsets from OpenAudioBench (Li et al., 2025b), including AlpacaEval (Li et al., 2023), LLaMA Questions (Nachmani et al., 2023), TriviaQA (Joshi et al., 2017), and Web Questions (Berant et al., 2013). For the evaluation process, we employ Qwen-max [3] to score and determine the correctness of responses. All scores are scaled and normalized to a range of 0 to 10. Further details are provided in Appendix E.

Furthermore, we employ two metrics to evaluate the quality of the generated speech. To assess the overall speech quality, we use the UTMOS (Saeki et al., 2022) to predict mean opinion scores (MOS). To assess the alignment between speech and text responses, we transcribe the speech by Whisper-large-v3 (Radford et al., 2023) and compute the word error rate (WER) between the transcription and the corresponding text response.

5 Experiments Results

5.1 Overall Result

Table 1 presents the performance of VocalNet in voice assistant scenario compared to other mainstream speech LLMs and omni LLMs. For all evaluated models, we input speech queries and require the models to generate both speech and text responses simultaneously, which are then assessed separately. For the $s \rightarrow t$ modality, we evaluate the text response directly, while for the $s \rightarrow s$ modality, we first transcribe the speech response using Whisper-large-v3 before conducting the evaluation.

For small-scale speech LLMs (LLM size $\leq$ 1B), VocalNet-1B significantly outperforms Mini-Omni and SLAM-Omni, both based on Qwen2-0.5B. Despite having roughly twice the parameter count, VocalNet-1B achieves substantial improvements—for instance, 71.7% accuracy on LLaMA Questions, compared to 2.7% and 29.4% for Mini-Omni and SLAM-Omni respectively. Notably, VocalNet-1B even surpasses several base-sized models ($\sim$8B) on specific subsets. On AlpacaE-

[3] https://qwenlm.github.io/blog/qwen2.5-max/

Model	LLM size	Modality	AlpacaEval	LLaMA Q.	TriviaQA	Web Q.	Avg. Score
Mini-Omni	0.5B	s→t	1.84	0.27	0.12	0.22	0.61
		s→s	1.80	0.27	0.08	0.20	0.59
SLAM-Omni	0.5B	s→t	3.50	2.94	0.39	0.84	1.92
		s→s	3.01	2.67	0.34	0.69	1.68
VocalNet-1B	1B	s→t	**5.79**	**7.17**	**3.60**	**5.16**	**5.43**
		s→s	**5.03**	**6.37**	**3.06**	**4.68**	**4.79**
LLaMA-Omni	8B	s→t	5.31	6.97	4.44	5.44	5.54
		s→s	3.89	5.51	2.44	4.00	3.96
Freeze-Omni	7B	s→t	4.51	7.77	5.32	6.41	6.00
		s→s	2.99	6.02	3.53	4.78	4.33
GLM-4-Voice	9B	s→t	5.86	7.74	4.95	5.56	6.03
		s→s	5.27	6.43	4.63	5.40	5.43
Baichuan-Omni-1.5	7B	s→t	5.20	7.76	5.72	6.12	6.20
		s→s	4.10	6.12	4.13	5.18	4.88
MiniCPM-o	8B	s→t	6.13	7.72	**6.43**	**7.16**	6.86
		s→s	4.95	6.58	4.99	6.22	5.69
Qwen2.5-Omni	8B	s→t	6.01	7.90	5.89	6.88	6.67
		s→s	5.73	**7.63**	5.59	**6.70**	**6.41**
VocalNet-8B	8B	s→t	**7.12**	**7.95**	6.24	6.48	**6.95**
		s→s	**6.37**	7.31	**5.67**	6.16	6.38

Table 1: Comparison with different speech LLMs and omni LLMs on OpenAudioBench. **Bold** indicates the optimal result in each subgroup.

val, it outperforms LLaMA-Omni, Freeze-Omni, and Baichuan-Omni-1.5; on LLaMA Questions, it exceeds LLaMA-Omni. VocalNet-8B achieves performance on par with MiniCPM-o and Qwen2.5-Omni, and consistently outperforms other base-sized models. It ranks the top two on AlpacaEval, LLaMA Questions, and TriviaQA in both $s \rightarrow t$ and $s \rightarrow s$ modalities. On Web Questions, it places third, slightly behind MiniCPM-o and Qwen2.5-Omni, demonstrating strong overall performance across the evaluated models.

To assess the quality of generated speech, we report the average WER and UTMOS scores. As shown in Table 2, VocalNet-1B outperforms all other small-scale models across all metrics. VocalNet-8B preserves its advantage in speech fidelity and achieves the second-lowest WER, slightly behind only Qwen2.5-Omni.

Model	WER↓	UTMOS↑
Mini-Omni	8.66	4.43
SLAM-Omni	6.17	4.46
VocalNet-1B	**5.31**	**4.49**
LLaMA-Omni	15.90	3.96
Freeze-Omni	18.31	4.40
GLM-4-Voice	8.99	4.23
Baichuan-Omni-1.5	22.67	4.35
MiniCPM-o	8.72	4.14
Qwen2.5-Omni	**2.63**	4.34
VocalNet-8B	3.56	**4.49**

Table 2: Comparison with different models in generated speech quality. **Bold** indicates the optimal result in each subgroup.

5.2 Training Strategy

Previous speech LLMs often adopt a multi-stage training pipeline that includes ASR and TTS pre-training. However, the necessity of these pre-training stages has not been sufficiently validated, and they introduce additional computational costs. To investigate their effectiveness, we conducted experiments with and without ASR and TTS pre-training. As shown in Table 3, ASR pre-training demonstrated minimal impact on model performance, which remained close to that of the model without any pre-training. While TTS pre-training improved multi-modal alignment, evidenced by a lower WER, it substantially degraded the model's scores on OpenAudioBench, particularly the average score (s2s), which dropped from 5.43 to 4.99. These results suggest that neither ASR nor TTS pre-training provides a substantial improvement to overall model capabilities, while both contribute to computational overhead. Therefore, for simplicity and efficiency, we have removed both stages from our final training framework.

We further explore whether the two stages described in Section 2.2 can be merged into a sin-

Setting	Score(s2t)↑	Score(s2s)↑	WER↓	UTMOS↑
VocalNet-1B	5.43	**4.79**	5.31	4.49
-w / ASR pre-training	**5.48**	4.65	6.18	4.49
-w / TTS pre-training	4.99	4.62	**4.92**	**4.50**

Table 3: Ablation study on the effect of ASR and TTS pre-training. *s2t* denotes $s \rightarrow t$ and *s2s* denotes $s \rightarrow s$.

gle unified training phase. As shown in Table 4, the one-stage approach achieves comparable performance for VocalNet-1B. Although it leads to a slight drop in spoken QA tasks under the $s \rightarrow t$ setting, it yields a marginally lower WER. Considering the overall performance in spoken QA tasks and greater flexibility for speech generation experiments in Section 5.3, we ultimately adopt the two-stage training framework. More detailed experimental setups are provided in Appendix D.

Setting	Score(s2t)↑	Score(s2s)↑	WER↓	UTMOS↑
One-stage	5.21	4.78	**5.18**	4.48
Two-stage	**5.43**	**4.79**	5.31	**4.49**

Table 4: Performance comparison between one-stage and two-stage training strategies for VocalNet-1B.

5.3 MTP Implementation

In this section, we conduct experiments with the five MTP implementations discussed in Section 3.2, utilizing the LLaMA-3.2-1B-Instruct as the backbone and trained with the VoiceAssistant-400K dataset. Results are shown in Table 5. Group-linear and Group-Trans denote the group modeling approaches employed in SLAM-omni and IntrinsicVoice respectively. We test the group sizes of 3 and 5. The results show that while group modeling can improve the generation speed of speech tokens, it leads to a decline compared to NTP. This is especially noticeable with a larger group size, where both metrics exhibit considerable deterioration.

For the other MTP implementations, the number of tokens predicted per step can be flexibly adjusted during inference. In this study, we fix the number of MTP modules to 5 during training and evaluate performance when predicting 1, 3, and 5 tokens per step during inference. For MTP-Parallel-Linear, the use of parallel linear layers disrupts the temporal dependencies among speech tokens, leading to a noticeable degradation in both WER and UTMOS as more tokens are predicted per step. This suggests that without explicit modeling of inter-token temporal dependencies, the quality of generated speech deteriorates significantly when predicting a larger

Method	G.S. / M.N.	Tokens per Step	WER↓	UTMOS↑
Baseline(NTP)	-	1	10.62	4.488
Group-Linear	3	3	11.50	4.488
	5	5	17.61	4.414
Group-Trans	3	3	14.34	4.489
	5	5	17.90	4.468
MTP-Parallel-Linear	5	1	8.61	4.492
		3	8.00	4.494
		5	10.57	4.467
MTP-DeepSeek	5	1	9.14	4.493
		3	9.02	**4.498**
		5	18.23	4.488
MTP-VocalNet	5	1	6.84	4.494
		3	**5.66**	4.495
		5	6.46	4.486

Table 5: Comparison of different MTP implementations. G.S.: Group Size; M.N.: MTP Module Number during Training. Tokens per Step: Number of tokens predicted per inference step. **Bold** indicates the best result.

number of tokens simultaneously. Similarly, MTP-DeepSeek exhibits a substantial performance drop when predicting 5 tokens per step. This decline is likely due to the teacher-forcing next-step prediction strategy used during training, as discussed in Section 3.2.

In contrast to these approaches, our proposed architecture demonstrates superior performance. Notably, even when predicting 5 tokens per step, our method maintains a high UTMOS score and an exceptionally low WER. These results strongly validate the effectiveness of our MTP implementation, as it successfully addresses the limitations observed in previous methods. We attribute the improvements in speech generation brought by the MTP paradigm to two key factors: reduced error accumulation in autoregressive modeling and enhanced modeling of local speech patterns and temporal dependencies. A detailed analysis and validation of these aspects are provided in Appendix C.

6 Conclusion

We present VocalNet, a series of advanced speech LLMs overcoming key efficiency and quality challenges through two innovations. First, a streamlined two-stage training efficiently integrates speech capabilities into pre-trained LLMs. Second, multi-token prediction (MTP) offers a superior alternative to autoregressive speech generation, achieving faster inference and enhanced quality. Experiments show VocalNet rivals leading Omni models (e.g., MiniCPM-o, Qwen2.5-Omni) on OpenAudioBench and markedly surpasses prior open-source speech LLMs in speech quality. These results affirm the efficacy of our methodology for developing high-performance speech LLMs.

Limitations

Our work has the following limitations. First, although VocalNet achieves strong performance trained on a limited amount of data, it currently lacks the capability for controllable speech generation and paralinguistic modeling. As a result, we plan to collect and incorporate more high-quality, diverse speech data in future work to enhance and explore these aspects. Second, VocalNet currently relies on the speech tokenizer from CosyVoice2 as the target for speech tokens. This choice may limit the model's ability in controllable speech generation and paralinguistic modeling, as semantic speech tokens are used. Furthermore, converting these speech tokens into audio requires a flow-matching model, which, according to results in the Appendix B, is the primary source of latency in VocalNet. Therefore, we also identify speech token design and further optimization of the speech decoder as key directions for future research.

Ethical Considerations

All pre-trained models used in this work were obtained from publicly available sources like HuggingFace and ModelScope. We strictly adhered to the respective license and usage terms associated with each model. No models were used outside the scope of their intended licenses. The datasets employed in our experiments are publicly available and were used in compliance with their specified licenses. We did not collect or curate any original data for this study. The speech data utilized in our experiments were either sourced from publicly available datasets or synthesized using open-source text-to-speech tools based on these datasets. Thereby, we minimized potential risks related to privacy, consent, and data misuse. By relying on established, ethically sourced data and avoiding any form of private or sensitive information, we ensured that our research adhered to responsible AI practices throughout the development and evaluation process.

References

Keyu An, Qian Chen, Chong Deng, Zhihao Du, Changfeng Gao, Zhifu Gao, Yue Gu, Ting He, Hangrui Hu, Kai Hu, and 1 others. 2024. Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms. *arXiv preprint arXiv:2407.04051*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. In *International Conference on Machine Learning*, pages 5209–5235. PMLR.

Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, and 1 others. 2025. Minmo: A multimodal large language model for seamless voice interaction. *arXiv preprint arXiv:2501.06282*.

Wenxi Chen, Ziyang Ma, Ruiqi Yan, Yuzhe Liang, Xiquan Li, Ruiyang Xu, Zhikang Niu, Yanqiao Zhu, Yifan Yang, Zhanxun Liu, and 1 others. 2024. Slam-omni: Timbre-controllable voice interaction system with single-stage training. *arXiv preprint arXiv:2412.15649*.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, and 1 others. 2024. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2024. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*.

Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Yangze Li, Zuwei Long, Heting Gao, Ke Li, and 1 others. 2025. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Roziere, David Lopez-Paz, and Gabriel Synnaeve. 2024. Better & faster large language models via multi-token prediction. In *International Conference on Machine Learning*, pages 15706–15734. PMLR.

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.

Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu,

Zhiqing Hong, Jiawei Huang, Jinglin Liu, and 1 others. 2024. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23802–23804.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, and 1 others. 2024. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033.

Bohan Li, Hankun Wang, Situo Zhang, Yiwei Guo, and Kai Yu. 2025a. Fast and high-quality autoregressive speech synthesis via speculative decoding. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Yadong Li, Jun Liu, Tao Zhang, Song Chen, Tianpeng Li, Zehuan Li, Lijun Liu, Lingfeng Ming, Guosheng Dong, Da Pan, and 1 others. 2025b. Baichuan-omni-1.5 technical report. *arXiv preprint arXiv:2501.15368*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024. Eagle: speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, pages 28935–28948.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Run Luo, Ting-En Lin, Haonan Zhang, Yuchuan Wu, Xiong Liu, Min Yang, Yongbin Li, Longze Chen, Jiaming Li, Lei Zhang, and 1 others. 2025. Openomni: Large language models pivot zero-shot omnimodal alignment across language with real-time self-aware emotional speech synthesis. *arXiv preprint arXiv:2501.04561*.

Kentaro Mitsui, Koh Mitsuda, Toshiaki Wakatsuki, Yukiya Hono, and Kei Sawada. 2024. Pslm: Parallel generation of text and speech with llms for low-latency spoken dialogue systems. *arXiv preprint arXiv:2406.12428*.

Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. 2023. Spoken question answering and speech continuation using spectrogram-powered llm. *arXiv preprint arXiv:2305.15255*.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, and 1 others. 2025. Spirit-lm: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52.

OpenAI. 2024. https://openai.com/index/hello-gpt-4o/.

OpenBMB. 2025. Minicpm-o 2.6: A gpt-4o level mllm for vision, speech, and multimodal live streaming on your phone. https://openbmb.notion.site/185ede1b7a558042b5d5e45e6b237da9. Accessed: 2025-03-28.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequencepre-training. *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *arXiv preprint arXiv:2204.02152*.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180.

Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. 2024. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*.

Zhifei Xie and Changqiao Wu. 2024. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. In *Proc. Interspeech 2019*, pages 1526–1530.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chaohong Tan, Zhihao Du, and 1 others. 2024a. Omniflatten: An end-to-end gpt model for seamless voice conversation. *arXiv preprint arXiv:2410.17799*.

Xin Zhang, Xiang Lyu, Zhihao Du, Qian Chen, Dong Zhang, Hangrui Hu, Chaohong Tan, Tianyu Zhao, Yuxuan Wang, Bin Zhang, and 1 others. 2024b. Intrinsicvoice: Empowering llms with intrinsic real-time voice interaction abilities. *arXiv preprint arXiv:2410.08035*.

A Related Work

A.1 End-to-End Speech Interaction System

End-to-end speech interaction systems have become a key research focus in the speech processing community. As discussed in Chen et al. (2025), speech LLMs can be categorized into two types: native multimodal models and aligned multimodal models. Native multimodal speech LLMs generate tokens for both modalities using a unified backbone. These models can be further divided into two categories: one type, represented by Mini-Omni (Xie and Wu, 2024), Moshi (Défossez et al., 2024), PSLM(Mitsui et al., 2024) and SLAM-Omni (Chen et al., 2024), adopts a multi-stream architecture that simultaneously generates audio and text outputs. The other type, including Omni-Flatten (Zhang et al., 2024a), GLM-4-Voice (Zeng et al., 2024), SpiRit LM (Nguyen et al., 2025) and Baichuan-Omni-1.5 (Li et al., 2025b), generates interleaved audio and text outputs to handle both modalities. However, these models require large amounts of speech-text pairs for training to avoid catastrophic forgetting. Even using a large amount of training data, their knowledge and reasoning capabilities often fall short compared to similar-sized LLMs.

Alternatively, aligned multimodal models introduce separate encoders, decoders, and vocoders for speech processing. This architecture has the advantage of preserving the original abilities of LLMs while also generating high-quality speech responses. LLaMA-Omni (Fang et al., 2024) uses a non-autoregressive method based on connectionist temporal classification (CTC) (Graves et al., 2006) for speech generation. Although it offers low latency, the quality of the generated speech is relatively poor. Freeze-Omni (Wang et al., 2024), MiniCPM-o (OpenBMB, 2025), MinMo (Chen et al., 2025) and VITA-1.5 (Fu et al., 2025) all employ autoregressive speech decoders trained with the next-token prediction task for speech generation. Qwen2.5-Omni (Xu et al., 2025) introduces a dual-track autoregressive Transformer decoder architecture for speech decoding, which enables more natural streaming inference without modifying the training process. However, the superiority of this dual-stream framework in speech modeling still requires further investigation in future research.

A.2 Multi-token Prediction

Multi-token prediction has emerged as an important advancement in language modeling, offering improvements in sample efficiency, reasoning capabilities, and inference speed. The concept of multi-token prediction was initially explored by Qi et al. (2020), who proposed training models to predict several future tokens in parallel. Building upon this foundation, Gloeckle et al. (2024) introduced a refined architecture that incorporated multiple output heads operating over a shared model backbone. Their approach demonstrated that multi-token prediction could lead to models that are both better and faster. Furthermore, Cai et al. (2024) proposed a speculative decoding method based on multi-token prediction to accelerate LLM inference.

In the context of speech generation, several works have employed group modeling techniques to implement multi-token prediction. SLAM-Omni (Chen et al., 2024) proposes a semantic group modeling approach to accelerate speech to-

ken generation and model training. This method partitions the speech token sequence into fixed-size groups and uses a linear layer to reconstruct each group embedding into multiple speech tokens. Similarly, IntrinsicVoice (Zhang et al., 2024b) introduces GroupFormer, a non-autoregressive Transformer module to perform token reconstruction. While group modeling methods can accelerate speech generation, they often lead to quality degradation, particularly as the group size increases.

B Streaming Speech Decoding

B.1 Attention Mask Design

To enable efficient speech decoding in streaming scenarios while ensuring high-quality non-streaming speech decoding, we employ two attention mask mechanisms tailored for complete sequence processing and real-time speech generation respectively, inspired by (OpenBMB, 2025). During the generative supervised fine-tuning stage, these two mask mechanisms are used simultaneously in a batch, allowing the model to flexibly adapt to diverse decoding requirements.

Non-Streaming Attention Mask The non-streaming attention mask as shown in Figure 3 (a), is optimized for scenarios involving the one-time processing of complete input sequences. BOS and SOS refer to 'begin of stream' and 'switch of stream', two identified special tokens. The yellow blocks refer to the attended text positions during speech generation, and the blue and red ones are the attended positions within the same modality. In this mode, the text hidden states $\boldsymbol{v}_{LLM}$ generated by the speech projector from $\boldsymbol{h}_{LLM}$ are fully visible to themselves, while the attention for the speech component adheres to an autoregressive property, meaning each speech token $\boldsymbol{s}^i$ depends solely on itself and preceding tokens. Additionally, speech tokens $\boldsymbol{s}^i$ have unrestricted access to the text hidden states $\boldsymbol{v}_{LLM}$, leveraging global contextual information comprehensively.

Given the text hidden state $\boldsymbol{v}_{LLM} \in \mathbb{R}^{L_t}$ with length L_t and the speech hidden state $\boldsymbol{s} \in \mathbb{R}^{L_s}$ with length L_s, the attention mask $\boldsymbol{A} \in \{0,1\}^{(L_t+L_s)\times(L_t+L_s)}$ for a single instance is defined:

$$A_{i,j} = \begin{cases} 1 & i \leq L_t \\ 1 & i > L_t, i \geq j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Streaming Attention Mask The streaming attention mask as shown in Figure 3 (b), is specifically designed for real-time speech generation, supporting the incremental processing of input sequences. In this mode, both the text hidden states $\boldsymbol{v}_{LLM}$ and speech hidden states $\boldsymbol{s}$ are constrained by an autoregressive mask, permitting access only to preceding positions.

Let the speech sequence length L_s be divided into chunks of length C_s, with each along with increased visible real text positions (excluding BOS token) of length C_t. In Figure 3 (b), C_s and C_t is shown as 6 and 3 respectively. The streaming mask is formally defined as follows:

$$A_{i,j} = \begin{cases} 1 & i \leq L_t, i \geq j \\ 1 & i > L_t, i \geq j > L_t \\ 1 & i > L_t, j \leq min(L_t, \\ & \lceil (i - L_t - 1)/C_s \rceil \cdot C_t + 1) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

B.2 Performance Analysis

To provide a comprehensive evaluation of VocalNet, we conduct both latency analysis and performance comparison between streaming and non-streaming decoding modes. For speech generation, we measure the latency from receiving the speech input to producing the first chunk of generated speech response. As shown in Table 6, the speech response delay is broken down into four stages: (1) speech query encoding via Whisper, (2) LLM hidden state generation, (3) speech token prediction by the decoder, and (4) waveform construction using the vocoder. The latency for the LLM and speech decoder is measured based on generating 5 text tokens($C_t = 5$) and 15 speech tokens($C_s = 15$), with the MTP decoder predicting 3 speech tokens per step. We evaluate the model's latency on the LLaMA Questions dataset and report the average results.

The overall latency of VocalNet-1B and VocalNet-8B is approximately 320 ms and 430 ms, respectively. Notably, over half of this delay stems from the speech vocoder, particularly during the flow-matching model. All measurements are conducted on a single L20 GPU.

In addition to latency analysis, we compare the model's performance in both streaming and non-streaming modes, as shown in Table 7. Experimental results indicate that while streaming mode introduces some degradation in multi-modal alignment

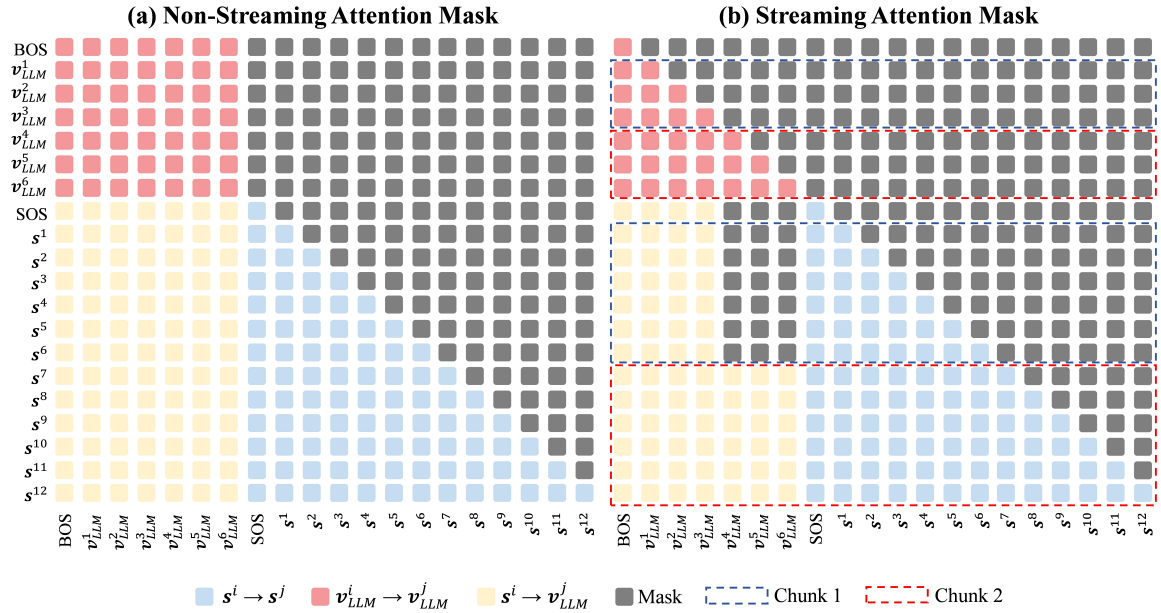

Figure 3: (a) Non-Streaming Attention Mask: v^i_{LLM} attends to itself and all text positions, and s^i attends to itself, all text positions, and its previous speech positions; (b) Streaming Attention Mask: v^i_{LLM} attends to itself and its previous text positions, and s^i attends to itself, chunk-limited text positions, and its previous speech positions.

Model	Speech Encoder (ms)	LLM (ms)	Speech Decoder (ms)	Speech Vocoder (ms)	Sum (ms)
VocalNet-1B	35.86	33.95	24.74	225.18	319.73
VocalNet-8B	36.08	126.71	40.02	225.56	428.38

Table 6: Speech generation latency of VocalNet. Experiments are conducted on 1 NVIDIA L20 GPU.

Model	Streaming	Avg. Score(s2t)↑	Avg. Score(s2s)↑	Avg. WER↓	Avg. UTMOS↑
VocalNet-1B	×	5.43	4.79	5.31	4.49
	✓	5.43	4.70	7.18	4.41
VocalNet-8B	×	6.95	6.38	3.56	4.49
	✓	6.95	6.33	5.95	4.40

Table 7: Comparison of streaming vs. non-streaming modes in VocalNet.

and speech quality, the impact remains relatively small. Specifically, the average scores of s2s decrease slightly under streaming (e.g., 4.79 vs. 4.70 for VocalNet-1B, and 6.38 vs. 6.33 for VocalNet-8B), suggesting that the model maintains strong voice interaction capability even under real-time scenario. Meanwhile, speech quality, as reflected by WER and UTMOS, experiences a moderate drop in streaming mode, but overall performance remains acceptable.

C Supplement to Multi-Token Prediction

In this section, we provide an in-depth analysis of the role of multi-token prediction (MTP) in speech generation from two perspectives: its effectiveness in mitigating error accumulation and its benefits in helping the model learn local speech characteristics. Furthermore, we conduct ablation studies on the configuration of MTP modules, investigating how the number of modules impacts overall performance.

C.1 Analysis of the Impact of MTP in Speech Generation

C.1.1 Mitigating Error Accumulation

Autoregressive models are commonly trained using teacher forcing, where the model is provided with the correct history tokens as input during training. However, during inference, the model generates outputs based on the predicted history in the autoregressive manner, which leads to the accumulation of errors. In speech generation tasks, we observe that the multinomial distributions predicted by our model tend to exhibit a flattened pattern. Figure 4 illustrates the distribution of maximum probabilities and entropy values across 70k predicted speech token distributions from VocalNet-1B trained with the NTP task. The results show that the maximum probabilities predominantly cluster below 0.25, while the entropy values generally exceed 3. Our observation indicates that most of the speech predictions contain multiple tokens with

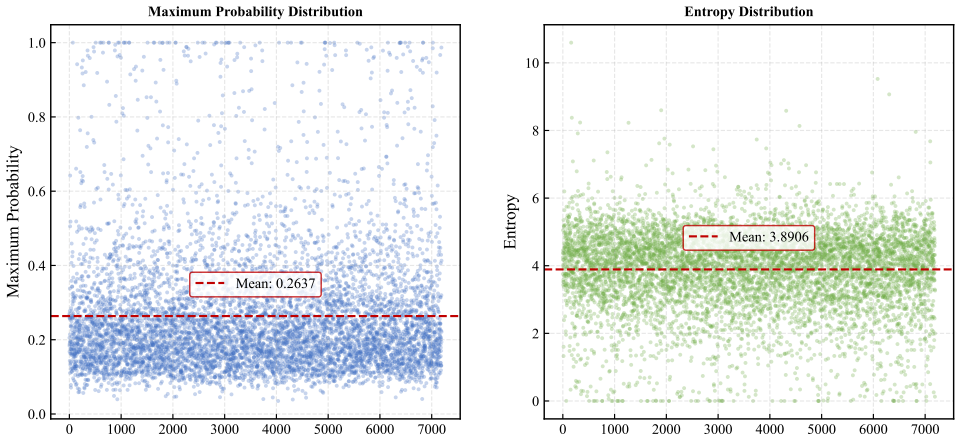

Figure 4: Distribution of maximum probabilities and entropy values for 70k predicted speech tokens from VocalNet-1B, trained with the NTP task. Red dashed lines represent the means.

similar probabilities, reflecting high uncertainty in the model's predictions. This phenomenon contributes to the worsening of error accumulation during speech generation. With an MTP loss added to the model training, this issue could be mitigated. The MTP loss is expressed as follows:

$$\begin{aligned}\mathcal{L}_{\text{MTP}} &= -\sum_{\boldsymbol{x}} \log q(\boldsymbol{x}_{t+1:t+K}|\boldsymbol{x}_{\leq t}), \\ &= -\sum_{\boldsymbol{x}} \sum_{k} \log q(\boldsymbol{x}_{t+k}|\boldsymbol{x}_{\leq t}),\end{aligned} \tag{7}$$

where q denotes the model's predictions, t represents the current time step, $\boldsymbol{x}$ refers to the data sample, $\boldsymbol{x}_{\leq t}$ denotes the historical sequence up to time t, and $K > 1$ indicates the number of future steps that need to be predicted.

As shown in Equation 7, the MTP loss function compels the model to learn to generate the correct future tokens $\boldsymbol{x}_{t+k}$ based on incomplete history $\boldsymbol{x}_{<t}$. This strategy allows the model to better handle the inherent uncertainty in the autoregressive process, leading to more accurate and robust predictions even when faced with noisy input history. As a result, the model becomes less dependent on perfect target sequences and more resilient to the noise introduced during inference.

C.1.2 Effectively Capturing Local Patterns in Speech

The MTP loss, by directly learning the joint distribution $p(\boldsymbol{x}_{t+1:t+k}|\boldsymbol{x}_{<t})$ of speech tokens, encourages the model to capture short-term temporal relationships and understand the underlying local dependencies within speech. In practice, multiple MTP modules can generate predictions for several future tokens based on the hidden state of the final layer of the speech decoder. This setup enables the model to anticipate the potential impact of future tokens while predicting the current token, effectively modeling local dependencies between them.

From an information-theoretic perspective, Gloeckle et al. (2024) demonstrate that in a two-token prediction setting, the MTP loss increases the weight of relative mutual information in a loss decomposition, which aids the model in better capturing the local relationship between adjacent tokens. Specifically, let $p(\cdot)$ denote the true data distribution and $q(\cdot)$ represent the densities of the model's predictions. Let $D(p||q)$ be the Kullback-Leibler divergence from q to p, and $H(p, q)$ be the cross-entropy from q to p. Gloeckle et al. (2024) show that the NTP loss can be decomposed as:

$$\begin{aligned}H(p_X, q_X) = H(p_{X|Y}, q_{X|Y}) \\ + I_{p||q}(X;Y)\end{aligned} \tag{8}$$

where X denotes the current token and Y denotes the second-next token, with conditioning on the preceding context C omitted for notational simplicity. The relative mutual information $I_{p||q}(X;Y)$ of X and Y from q relative to p is defined as:

$$I_{p||q}(X;Y) = D(p||q_X \otimes q_Y) - D(p||q) \tag{9}$$

Accordingly, the MTP loss can be expressed as:

$$\begin{aligned}H(p_X, q_X) + H(p_Y, q_Y) = H(p_{X|Y}, q_{X|Y}) \\ +2I_{p||q}(X;Y) + H(p_{Y|X}, q_{Y|X})\end{aligned} \tag{10}$$

Here, $H(p_{Y|X}, q_{Y|X})$ corresponds to the next-step NTP loss. Compared to NTP, MTP introduces an additional term $H(p_{Y|X}, q_{Y|X})$ and doubles the weight of the relative mutual information

Module Num	Tokens per Step	AlpacaEval		LLaMA Questions		TriviaQA		Web Questions		Avg	
		WER	UTMOS	WER	UTMOS	WER	UTMOS	WER	UTMOS	WER	UTMOS
3	1	5.38	4.489	5.24	**4.504**	7.59	**4.500**	9.23	4.484	7.79	4.493
	3	**3.37**	4.493	3.95	4.498	5.97	4.498	6.43	4.485	5.70	4.493
5	1	4.14	4.485	4.48	4.502	6.52	4.497	8.41	4.491	6.84	4.495
	3	3.43	**4.495**	**3.65**	4.498	5.97	4.499	**6.40**	4.489	**5.66**	4.495
	5	3.84	4.478	4.28	4.493	6.40	4.489	7.70	4.483	6.46	4.486
7	1	5.38	4.489	5.24	4.502	7.59	4.480	9.23	4.490	7.79	4.487
	3	3.40	4.490	3.92	4.499	**5.91**	4.498	7.57	**4.494**	6.14	**4.496**
	5	4.26	4.481	4.33	4.489	6.32	4.496	8.76	4.484	6.89	4.489
	7	5.50	4.470	5.19	4.474	8.28	4.478	9.20	4.462	8.06	4.470

Table 8: Comparison of performance using different numbers of MTP modules (Module Num) during training, evaluated with varying numbers of tokens predicted per inference step (**Tokens per Step**). **Bold** indicates the optimal result and underline indicates the sub-optimal result.

term $I_{p||q}(X;Y)$. On one hand, this additional term implies that the MTP paradigm makes more efficient use of the training data—particularly beneficial when data is limited. On the other hand, by placing greater emphasis on relative mutual information, the model can more effectively exploit the mutual information between adjacent tokens under the true data distribution p, thereby enhancing its predictive capability and ability to capture subtle interdependencies. This is especially crucial in speech modeling, where understanding such local structures significantly improves predictive accuracy.

Local patterns are particularly important in speech modeling. Neighboring speech tokens typically correspond to related linguistic units, such as phonemes or syllables. If the model fails to adequately learn these patterns, it is prone to making pronunciation errors during speech generation. As shown in the Table 9, we further decompose the WER of VocalNet-1B trained with NTP and MTP on the VoiceAssistant-400K dataset. For MTP-VocalNet, we use the best configuration shown in Table 5, which involves training with five MTP modules and predicting 3 tokens per step during inference. The results indicate that the majority of errors stem from substitutions, while insertions and deletions remain minimal. This suggests that the primary issue with the NTP-trained model lies in incorrect pronunciations, rather than over- or under-generating speech segments. With our MTP implementation, substitution errors are significantly reduced, indicating that the model achieves a better understanding of local speech structures. Understanding these relationships is vital for maintaining coherence and rhythm in speech. By encouraging the model to capture these local dependencies, the MTP loss enhances its ability to generate speech that is not only contextually accurate but also naturally fluent. In this way, the MTP loss plays a crucial role in helping the model learn short-term dependencies, enabling it to more effectively handle the complex temporal structures that characterize natural speech.

Method	Substitutions	Insertions	Deletions	Overall WER
Baseline(NTP)	6.71	2.38	1.53	10.62
MTP-VocalNet	3.92	1.01	0.73	5.66

Table 9: WER breakdown by error type for VocalNet-1B trained under NTP and MTP objectives.

C.2 Limitations of MTP-Deepseek

While MTP-Deepseek, as described in Section 3.2, is capable of performing multi-token prediction during inference, its training methodology is essentially the same as that of standard next-token prediction (NTP). This is because the method feeds the ground truth token $x_{\leq i+k}$ into the k-th MTP module to predict x_{i+k+1}, and computes the loss based on teacher-forced one-step predictions:

$$\mathcal{L}_{\text{MTP-Deepseek}} = -\sum_{\boldsymbol{x}} \sum_{k} \log q(\boldsymbol{x}_{t+k+1} | \boldsymbol{x}_{\leq t+k}), \tag{11}$$

This formulation is identical to the standard NTP training objective. Consequently, the analyses presented in Section C.1 regarding mitigating error accumulation and effectively capturing local patterns in speech do not apply to MTP-Deepseek.

C.3 Ablation Study for MTP

To determine the optimal configuration for MTP modules, we conduct ablation studies on the number of MTP modules, as detailed in Table 8. The results indicate that the number of tokens predicted per inference step primarily affects modality alignment performance, with the best results typically

achieved when predicting 3 tokens per step. Acoustic performance remains high and only slightly decreases as more tokens are predicted per step (e.g., 5 or 7). Overall, the number of MTP modules used during training has a relatively small impact, with the best performance achieved when training with 5 modules and inferring 3 tokens per step. The results of VocalNet in Section 5.1 are also based on this configuration.

D Details of the Ablation Study on Training Strategy

This section provides detailed descriptions of the ablation study introduced in Section 5.2, focusing on the implementation of ASR and TTS pre-training, as well as the one-stage training approach.

For ASR pre-training, we conduct experiments using the LibriSpeech (Panayotov et al., 2015) dataset. We first reformulate the ASR data into instruction-following format, as illustrated in the Figure 5. During this stage, only the Downsample Adaptor is trained, while other components remain frozen. Afterward, we proceed with the standard two-stage training process as shown in Figure 1.

Reformulated ASR Data

Query: “<speech> \n Please transcribe the speech to text.”
Response: “The Duchess Josiana towards seventeen oh five, although Lady Josiana was twenty-three and Lord David forty-four, the wedding had not yet taken place, and that for the best reasons in the world.”

Figure 5: ASR Data Format.

For TTS pre-training , we introduce an additional pre-training phase between Stage 1 and Stage 2 to provide a better initialization for the speech decoder. We use the LibriTTS dataset (Zen et al., 2019), reformatted into the instruction format shown in Figure 6. Speech tokens are extracted using the tokenizer from CosyVoice2. During this stage, we train the speech projector, speech decoder, and LLM backbone with LoRA. After TTS pre-training is completed, we proceed to Stage 2 training. Notably, and in contrast to the methodology described in Section 2.2, the LLM backbone is also trained with LoRA during this Stage 2.

For the one-stage approach , we use the same training data as in the two-stage framework but merge both stages into a single training phase. In this setup, we jointly fine-tune the Downsample Adaptor, LLM backbone (with LoRA), speech projector, and speech decoder throughout the entire training process.

Reformulated TTS Data

Query: “Please repeat after me: Haughty, inaccessible, and audacious, he addressed sonnets to her, which Josiana sometimes read.”
Response: “Haughty, inaccessible, and audacious, he addressed sonnets to her, which Josiana sometimes read.”

Figure 6: TTS Data Format.

E Evaluation Details

Prompts for AlpacaEval

[Instruction]
Please act as an impartial judge and evaluate the quality of the response provided by a voice assistant to the user's question displayed below. Your evaluation should consider factors such as clarity, helpfulness, relevance, accuracy, conciseness, and ease of understanding. Begin your evaluation by providing a short explanation of your reasoning. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: '[rating]', for example: 'Rating: [[5]]'.
[Question]
{instruction}
[The Start of Assistant’s Answer]
{response_txt}
[The End of Assistant’s Answer]

Figure 7: Prompt for AlpacaEval.

To assess model performance, we employ the LLM-based evaluation approach. We use the evaluation prompts from OpenAudioBench (Li et al., 2025b) to evaluate both open-ended and semi-open QA tasks, including AlpacaEval (open-ended), and LLaMA Questions, TriviaQA, and Web Questions (semi-open). For semi-open QA tasks, reference answers are included in the evaluation prompt to

Prompts for TriviaQA and Web Questions

Your will be given a question, the reference answers to that question, and an answer to be judged. Your tasks is to judge whether the answer to be judged is correct, given the question and reference answers. An answer considered correct expresses or contains the same meaning as at least **one of** the reference answers. The format and the tone of the response does not matter.

You should respond in JSON format. First provide a one-sentence concise analysis for the judgement in field ‘analysis‘, then your judgment in field ‘judgment‘. For example,
'''json
{{"analysis": "<a one-sentence concise analysis for the judgement>", "judgment": < your final judgment, "correct" or "incorrect">}}
'''

Question
{instruction}

Reference Answer
{targets}

Answer To Be Judged
{answer_to_be_judged_text}

Figure 8: Prompt for TriviaQ and Web Questions.

assist the LLM in judging the correctness of the model’s responses. For open-ended tasks, and the LLM directly scores the responses based on multiple qualitative aspects such as relevance, clarity, and coherence. The detailed evaluation prompts are shown in Figure 7, 8 , and 9.

Prompts for LlaMA Questions

Background
You are a professional QA evaluation expert. You need to assess whether the model's answer is correct based on the standard answer.\n\n

Scoring Criteria
Correct: The answer matches or is equivalent to the standard answer \n
Incorrect: The answer is wrong or irrelevant to the question \n\n

Evaluation Guidelines
1. The expression of answers can be flexible, not requiring exact matches. For example: \n
  - Numbers can be expressed in either Arabic numerals or words \n
  - Proper nouns can be in either English or Chinese \n
  - Differences in punctuation can be ignored \n
2. Focus on whether the core meaning of the answer is correct \n

Output Format
Provide the reasoning for your score, then generate the result in "[]" format and make sure it contains "the score is [Correct]" or "the score is [Incorrect]", for example:
```
The answer is correct and equivalent to the standard answer, the score is [Correct]
```
or
```
The answer is incorrect and does not match the standard answer, the score is [Incorrect]
```
\n\n
Question:
{prompt}

Standard Answer:
{gt_answer}

Model's Answer:
{answer_text}

Figure 9: Prompt for LLaMA Questions.