From Text to Talk: Audio-Language Model Needs Non-Autoregressive Joint Training

Tianqiao Liu · Xueyi Li · Hao Wang · Haoxuan Li · Zhichao Chen · Weiqi Luo · Zitao Liu

Introduction

Current Problems In Audio-Language MLLM

Existing Audio-Language MLLMs generate interleaved text and audio tokens with a uniform AR objective, overlooking a fundamental asymmetry (Figure 1):

- **Dependency mismatch**: Text exhibits target-target dependencies where each token causally depends on its predecessors. Audio, however, is driven by source-target dependencies - tokens within a span **primarily** condition on the source text. Meanwhile, intra-span relation still matter for fluency and prosody, but do not require a fixed left-to-right order.
- **Error Propagation**: Forcing a strict left-to-right order on audio amplifies exposure bias; a single mis-predicted audio token cascades errors through the entire span, despite audio tokens being largely independent given the source text.

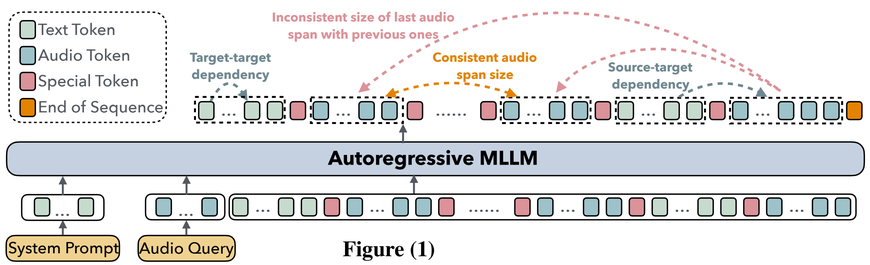

Figure (1)

Preliminary and Notation

- **Interleaved Sequence Notation**

We consider interleaved text-audio sequences with vocabulary $\mathbf{V} = \mathbf{V}_{\text{text}} \cup \mathbf{V}_{\text{audio}} \cup \mathbf{S}$, where $\mathbf{S} = \langle\text{SOA}\rangle, \langle\text{EOA}\rangle, \langle\text{EOS}\rangle, [\mathbf{M}]$ contains control and mask tokens. Each sequence alternates between text and audio spans: $x = (\mathbf{T}_1, \mathbf{A}_1, \ \ldots, \mathbf{T}_M, \mathbf{A}_M, \langle\text{EOS}\rangle)$ where $\mathbf{T}_m = (t_{m,1}, \ \ldots, t_{m,|\mathbf{T}_m|})$ and $\mathbf{A}_m = (a_{m,1}, \ \ldots, a_{m,|\mathbf{A}_m|})$, **A natural question is how to model this sequence. We begin with the standard autoregressive formulation.**

- **Autoregressive Modeling (AR)**

AR models factorize the joint probability via the chain rule $p(x) = \prod_{i=1}^{L} p(x^i \mid x^{<i})$. Applied to our interleaved setting $\mathscr{L}_{\text{AR}}(x) = -\sum_{m=1}^{M}\sum_{j=1}^{|\mathbf{T}_m|} \log\ p_\theta\big(t_{m,j} \mid \mathbf{T}_{<m}, \mathbf{A}_{<m}, t_{m,<j}\big)$

While AR suits text well, it imposes a rigid left-to-right order that **conflicts with the source-target nature of audio**. Discrete diffusion offers a flexible alternative.

- **Absorbing Discrete Diffusion = Any-Order AR**

Forward process: Each token is independently replaced with $[\mathbf{M}]$ with probability $1 - e^{-\bar{\sigma}(t)}$:

$$q(x_t^i \mid x_0^i) = \begin{cases} x_0^i & \text{with prob. } e^{-\bar{\sigma}(t)} \\ [\mathbf{M}] & \text{with prob. } 1 - e^{-\bar{\sigma}(t)} \end{cases}$$

Reverse process: The model learns to predict the original token at each masked position, while keeping visible tokens unchanged. A key result from Ou et al. (2024) shows that this objective is time-independent — the concrete score decomposes as:

$$\underbrace{\frac{p_t(\ldots, \hat{x}^i = v, \ \ldots)}{p_t(\ldots, x^i = [\mathbf{M}], \ \ldots)}}_{\text{concrete score}} = \underbrace{\frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}}}_{\text{time scalar}} \cdot \underbrace{p_0\big(v \mid x_{\text{vis}}\big)}_{\text{clean conditional}}$$

This is equivalent to an **any-order AR model (AO-ARM), averaged over all possible token orderings:**

$$\mathscr{L}_{\mathbf{AO}}(x) = \mathbb{E}_{\pi \sim U_\pi} \sum_{l=1}^{L} -\log\ q_\theta\big(x^{\pi(l)} \mid x^{\pi(<l)}\big)$$

Our Method - TtT

- **Core Idea**

TtT integrates AR text generation and NAR audio diffusion within a single Transformer initialized from a pretrained LLM. The model alternates between AR decoding and block-wise diffusion, controlled by special tokens $\langle\text{SOA}\rangle$ and $\langle\text{EOA}\rangle$. (Figure 2 & 3)

- **Partial-Order Formulation**

We formalize the dependency asymmetry as a partially ordered set (poset) $(V, \preceq)$:
i, Text tokens follow a total order: $t_{m,j} \preceq t_{m,j+1}$ (left-to-right causality).
ii, All tokens in span m preced ṣpan $m+1$ (cross-span dependency).
iii, Audio tokens within each span $\mathbf{A}_m$ form an anti-chain - no mandatory internal ordering.

This yields distinct predecessor sets:

$$\text{Pa}(t_{m,j}) = \mathbf{T}_{<m} \cup \mathbf{A}_{<m} \cup t_{m,<j} \qquad \text{Pa}(a_{m,j}) = \mathbf{T}_{\le m} \cup \mathbf{A}_{<m}$$

Text tokens depend on all prior tokens (target-target); audio tokens primarily depend on cross-modal context (source-target), while intra-span dependencies are naturally captured through the any-order AR nature of diffusion. We define the order-marginalized factorization of audio span:

$$\tilde{p}_\theta(\mathbf{A}_m \mid \mathbf{T}_{\le m}, \mathbf{A}_{<m}) = \mathbb{E}_{\pi_m \sim U_{\pi_m}} \prod_{j=1}^{|\mathbf{A}_m|} q_\theta\big(a_{m,\pi_m(j)} \mid \mathbf{T}_{\le m}, \mathbf{A}_{<m}, a_{m,\pi_m(<j)}\big)$$

where π_m is a random permutation over positions within span $\mathbf{A}m$, U_{π_m} is the uniform distribution over all such permutations, and $a_{m,\pi_m(<j)}$ denotes the audio tokens appearing before position j in the permuted order.

- **Unified Objective**

Combining fixed-order AR for text with **order-marginalized** any-order AR for audio:

$$\tilde{p}_\theta(x) = \prod_{m=1}^{M} \Bigg[\underbrace{\prod_{j=1}^{|\mathbf{T}_m|} p_\theta\big(t_{m,j} \mid \mathbf{T}_{<m}, \mathbf{A}_{<m}, t_{m,<j}\big)}_{\text{fixed-order AR for text}} \cdot \underbrace{\tilde{p}_\theta\big(\mathbf{A}_m \mid \mathbf{T}_{\le m}, \mathbf{A}_{<m}\big)}_{\text{order-marginalized for audio}} \Bigg]$$

Directly optimize $\tilde{p}_\theta(x)$ is intractable, because the order-marginalized conditional requires computing expectations over all permutations. By Jensen's inequality on the audio term:

$$\mathscr{L}_{\text{AO}}(x) = \mathbb{E}_{\pi_m \sim U_{\pi_m}} \sum_{j=1}^{|\mathbf{A}_m|} \Big[-\log\ q_\theta\big(a_{m,\pi_m(j)} \mid \mathbf{T}_{\le m}, \mathbf{A}_{<m}, a_{m,\pi_m(<j)}\big)\Big]$$

$$\ge\ -\log \mathbf{E}_{\pi_m \sim U_{\pi_m}} \prod_{j=1}^{|\mathbf{A}_m|} q_\theta\big(a_{m,\pi_m(j)} \mid \mathbf{T}_{\le m}, \mathbf{A}_{<m}, a_{m,\pi_m(<j)}\big) = \sum_{m=1}^{M} \Big(-\log \tilde{p}_\theta\big(\mathbf{A}_m \mid \mathbf{T}_{\le m}, \mathbf{A}_{<m}\big)\Big)$$

Thus:

$$\mathscr{L}_{\text{Unified}}(x) \ \triangleq\ \mathscr{L}_{\text{AR}}(x) + \mathscr{L}_{\text{AO}}(x) \ \ge\ \mathscr{L}_{\text{AR}}(x) + \sum_{m=1}^{M} \Big(-\log \tilde{p}_\theta\big(\mathbf{A}_m \mid \mathbf{T}_{\le m}, \mathbf{A}_{<m}\big)\Big) \ = -\log\ \tilde{p}_\theta(x)$$

$\mathscr{L}_{\text{Unified}}$ is a tractable upper bound on the NLL of the joint distribution $\tilde{p}_\theta(x)$, enabling standard gradient-based optimization.

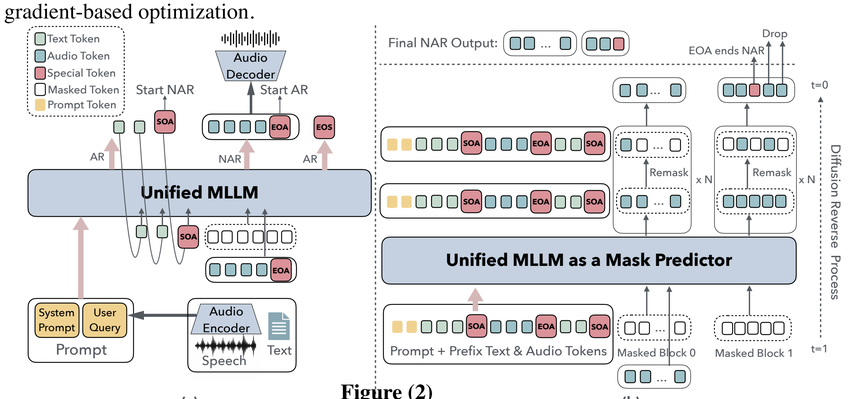

Figure (2)

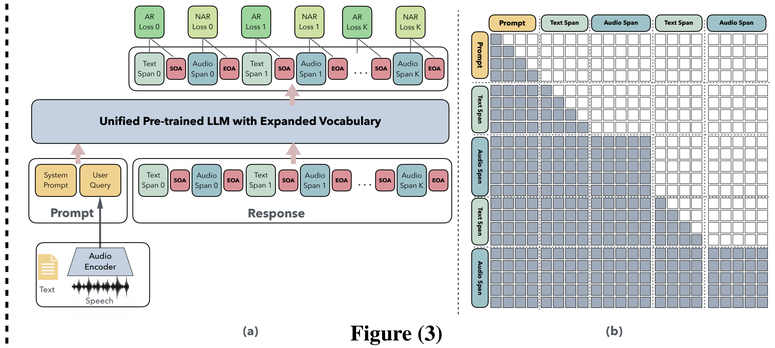

Figure (3)

- **Bridging Train-Test Discrepancy (Only one-forward pass during training)**

i, **BANOM**: With prob. p_{mix}, skip masking and train text with clean audio context.
ii, **PPM**: With prob. p_{prefix}, keep all spans before a random cutoff unmasked.
iii, **SST**: With prob. p_{trunc}, randomly truncate the final audio span to force content-aware $\langle\text{EOA}\rangle$ prediction.

Experiments

We evaluate against SOTA audio-language MLLMs on URO-Bench, Audio-QA, ASR, AAC, and S2S tasks, with perceptual quality assessment (NMOS & UTMOS) and ablation studies for each training strategy.

Table 1: Comprehensive evaluation of TtT framework. Higher (↑) is better for Audio-QA, lower (↓) is better for ASR. Datasets abbreviations are available in Table 7.

Models	Audio-QA (↑)				ASR (↓)					
	AE.	LQ.	TQA.	WQ.	Fzh.	A2.	A1.	WS_m.	WS_n.	Fen.
Main Results										
Qwen2.5-1.5B (AR)	10.85	1.00	0.00	0.10	103.18	81.84	95.96	103.15	95.54	105.62
Qwen2.5-1.5B (NAR)	10.70	0.00	0.40	0.20	86.97	224.37	191.11	123.96	143.76	108.25
TtT-1.5B (AR–NAR)	**15.68**	**23.75**	**3.47**	**7.70**	**44.36**	**14.89**	**16.72**	**52.23**	**41.52**	**49.00**
Qwen2.5-3B (AR)	14.42	10.00	0.60	0.70	90.32	54.94	72.01	80.01	73.64	74.47
Qwen2.5-3B (NAR)	11.31	0.67	1.21	0.70	68.94	212.27	160.58	89.22	111.29	83.51
TtT-3B (AR–NAR)	**17.46**	**34.68**	**6.53**	**11.61**	**55.67**	**12.53**	**13.65**	**53.83**	**44.29**	**64.31**
Ablation Study										
TtT-3B w/o BANOM	13.87	19.87	2.81	5.12	58.25	18.58	21.35	58.48	49.52	68.87
TtT-3B w/o PPM	14.27	22.79	2.71	5.54	58.86	15.63	18.83	57.76	47.92	67.37
TtT-3B w/o SST	14.12	10.20	1.30	3.72	56.39	25.43	31.03	64.41	56.70	**62.60**
TtT-3B (AR–NAR)	**17.46**	**34.68**	**6.53**	**11.61**	**55.67**	**12.53**	**13.65**	**53.83**	**44.29**	64.31
Training Strategy Comparison										
TtT-3B (AR–NAR)	17.46	34.68	6.53	11.61	55.67	12.53	13.65	53.83	44.29	64.31
Pretrain+AR	**29.45**	15.93	3.61	11.45	23.37	9.79	12.67	**26.75**	20.91	19.49
Pretrain+TtT	26.73	**40.07**	**11.07**	**21.43**	**18.99**	**6.80**	**5.78**	27.59	**19.85**	**19.10**

Table 2: Performance comparison on Audio-QA, ASR, and AAC tasks. Higher (↑) is better for Audio-QA and AAC; lower (↓) is better for ASR. Datasets abbreviations are available in Table 7.

Models	Size	Audio-QA (↑)				ASR (↓)						AAC (↑)	
		AE.	LQ.	TQA.	WQ.	Fzh.	A2.	A1.	WS_m.	WS_n.	Fen.	Clo.	MACS
Large Models (> 7B)													
Moshi	7B	25.63	48.30	16.75	16.85	-	-	-	-	-	-	4.32	12.01
SpeechGPT	7B	10.00	30.96	16.53	24.53	101.45	120.77	111.81	123.15	124.86	45.15	2.10	3.95
Kimi-Audio	7B	19.49	**57.53**	43.51	43.20	**2.87**	**2.53**	**0.61**	**6.34**	**5.39**	**4.87**	**55.92**	**64.90**
VITA-Audio	7B	40.20	54.30	18.59	30.75	6.35	5.56	4.58	20.38	15.88	9.58	6.18	7.94
LLaMA-Omni	8B	39.59	48.46	21.80	30.28	-	-	-	-	-	-	2.53	4.56
GLM-4-Voice	9B	**44.87**	**62.67**	**44.99**	**48.47**	-	-	-	-	-	-	13.15	12.67
Efficient Models (≤ 3B)													
Mini-Omni	0.5B	15.73	2.00	1.10	2.42	182.73	342.40	442.06	294.42	335.80	22.74	3.61	4.45
SLAM-Omni	0.5B	17.47	24.75	3.51	7.90	-	-	-	-	-	-	**54.52**	**50.46**
Qwen2.5-3B (AR)	3B	14.42	10.00	0.60	0.70	90.32	54.94	72.01	80.01	73.64	74.47	9.73	48.64
Qwen2.5-3B (NAR)	3B	11.31	0.67	1.21	0.70	68.94	212.27	160.58	89.22	111.29	83.51	9.54	27.40
TtT	3B	17.46	34.68	6.53	11.61	55.67	12.53	13.65	53.83	44.29	64.31	12.63	48.87
Pretrain+TtT	3B	**26.73**	**40.07**	**11.07**	**21.43**	**18.99**	**6.80**	**5.78**	**27.59**	**19.85**	**19.10**	11.55	42.86

Text and audio have fundamentally different dependency structures — respecting this asymmetry through hybrid AR-NAR training yields consistent gains. (We should rethink visual-language MLLMs)