Adaptive Condition Optimization for Text-to-Speech via Inference-Time Gradient Guidance

Anonymous Author(s)
Submission Id: 6080

Abstract

Existing zero-shot text-to-speech systems typically adopt a two-stage architecture combining a large language model with flow matching. However, flow matching follows an open-loop inference process without real-time correction during sampling, leading to speech distortion and unstable quality. A key limitation is that the condition is computed once before sampling and fixed throughout inference. Yet as the generation trajectory evolves, a static condition cannot adapt accordingly. To address this, we propose Adaptive Condition Optimization for Text-to-Speech (Aco-TTS), a gradient-guided framework for inference-time optimization that dynamically refines the condition during sampling using textual alignment and perceptual quality as guidance. This enables online correction and better constrains the flow trajectory. Aco-TTS is training-free and improves generation through gradient-based feedback at inference time. Experiments on SeedTTS, LibriSpeech, and AIShell-1 demonstrate significant reductions in word error rate and improvements in audio quality. Code and audio demonstrations are available at https://demo-tts.github.io.

CCS Concepts

• **Information systems** → **Multimedia content creation**; • **Computing methodologies** → *Neural networks.*

Keywords

Text-to-speech; Inference-time optimization; Gradient guidance

ACM Reference Format:
Anonymous Author(s). 202. Adaptive Condition Optimization for Text-to-Speech via Inference-Time Gradient Guidance. In *Proceedings of Proceedings of the 34th ACM International Conference on Multimedia (MM '26).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Recent zero-shot text-to-speech (TTS) systems have shown strong generalization to diverse speakers, languages, and contextual conditions [6, 18]. A representative line of work decouples condition feature modeling from continuous acoustic generation. Large pre-trained language or speech-language models are used to produce condition feature for downstream synthesis [11, 24, 30]. Flow matching is then used as the acoustic generator to recover continuous speech features from Gaussian noise [13, 18, 20, 21]. This design combines strong contextual representations with the high-fidelity generation capability of flow matching [20].

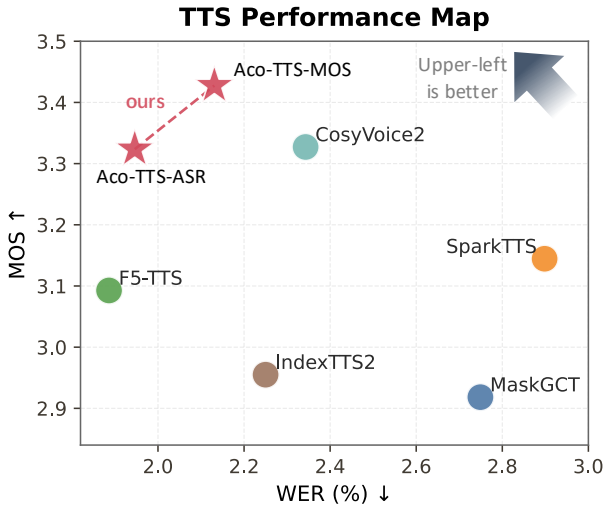

Figure 1: Performance comparison on the SeedTTS test-en benchmark, with the x-axis representing WER (lower is better) and the y-axis representing MOS (higher is better). Circles denote baseline systems, and stars denote our Aco-TTS variants. The dashed line connects the two Aco-TTS variants. Aco-TTS-ASR emphasizes intelligibility, while Aco-TTS-MOS emphasizes perceptual quality.

Despite these advances, the inference process of flow-matching-based TTS remains essentially open-loop, without explicit online correction during sampling [13, 20]. In challenging scenarios such as long-form synthesis or difficult pronunciations, zero-shot TTS systems can exhibit robustness failures, including mispronunciations, word skipping, and repetition [10, 16, 29]. Because sampling follows a deterministic and non-contractive probability-flow ODE, small errors can accumulate and be amplified along the generation trajectory [4]. These observations suggest that introducing online correction during inference is a promising direction for improving robustness [22].

A natural place to apply such correction is the condition feature [24]. In zero-shot TTS, conditioning information plays a central role in constraining the generation trajectory, carrying key information about linguistic content, speaker characteristics, and high-level acoustic attributes [11, 30]. However, in existing systems, condition features are computed once before sampling and kept fixed throughout inference [11, 23, 26, 30], making them difficult to adapt once the generation trajectory begins to drift. This limitation is further compounded by the fact that different diffusion or flow timesteps

MM '26, Rio de Janeiro, Brazil

ACM ISBN 978-1-4503-XXXX-X/202/06
https://doi.org/XXXXXXX.XXXXXXX

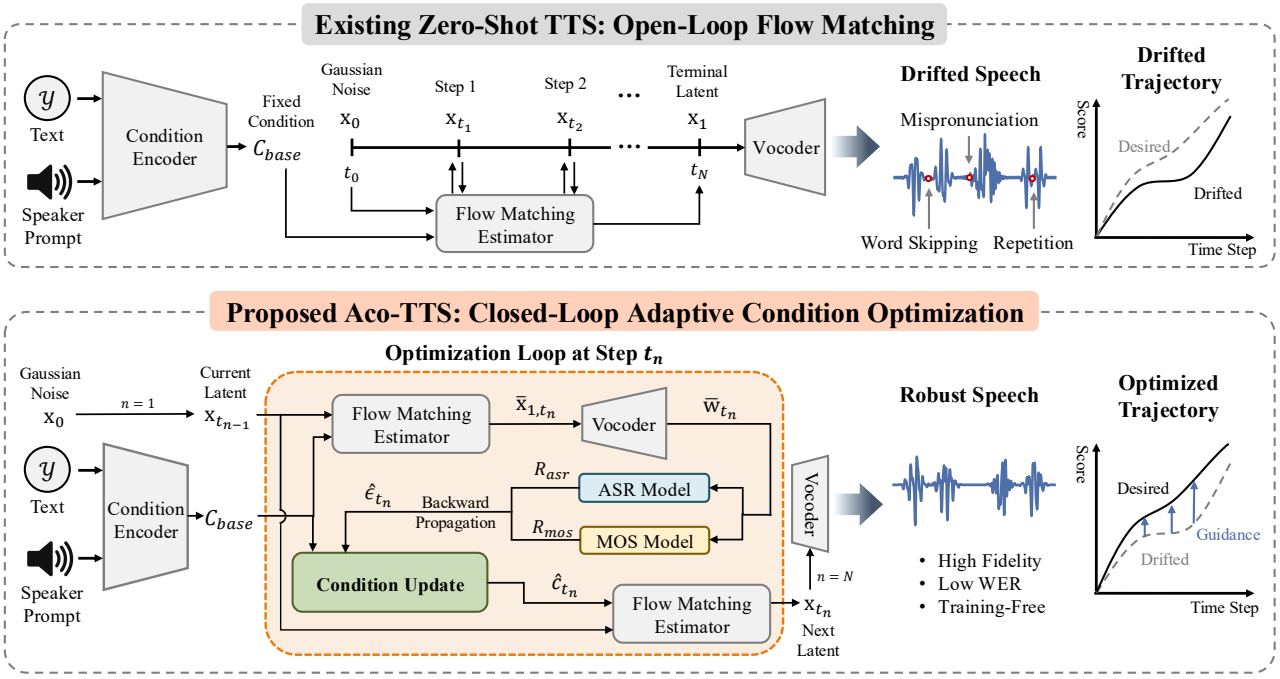

Figure 2: The overall framework of Aco-TTS. (Top) Existing open-loop flow-matching TTS systems often suffer from trajectory drifting and robustness issues. (Bottom) Our proposed closed-loop Aco-TTS dynamically optimizes the condition features $\mathbf{c}_t$ during inference using gradient guidance from differentiable ASR (FunASR [14]) and MOS (UTMOSv2 [2]) evaluators, ensuring high intelligibility and audio quality.

exert unequal influence on the generation process [8, 34], meaning a static condition is inherently mismatched to a generation process that evolves over time. This motivates dynamically updating the condition during inference rather than holding it fixed from start to finish.

Based on this motivation, we present Adaptive Condition Optimization for Text-to-Speech (Aco-TTS), an inference-time gradient guidance framework for TTS (see Figure 2). Aco-TTS introduces textual alignment and perceptual quality into the sampling loop through an ASR evaluator [14] and a MOS evaluator [2]. At each step, the current latent state is first used to predict an intermediate speech sample. This sample is then scored by the evaluator, and the resulting differentiable objective is backpropagated to the condition features. The updated condition is used in the next flow step to steer the trajectory toward more intelligible and higher-quality speech. In this way, Aco-TTS turns open-loop flow sampling into a closed-loop inference process with online correction. The whole procedure is training-free, does not modify model parameters, and can be plugged into existing cascaded TTS systems.

Experiments on SeedTTS, LibriSpeech, and AIShell-1 include main comparisons, step-size ablations, update-schedule studies, and reward-formulation analysis. As shown in Fig. 1, Aco-TTS consistently improves textual alignment and perceptual quality. These findings demonstrate that inference-time condition optimization is an effective and practical way to improve the robustness of zero-shot TTS, and that the proposed framework is both general and efficient.

The contributions of this work are summarized as follows:

- We propose Aco-TTS, an inference-time gradient guidance framework that improves the stability of zero-shot flow-matching TTS by dynamically optimizing condition features during sampling.
- We design a closed-loop optimization mechanism driven by textual alignment and perceptual quality signals, and show that backpropagating gradients into the condition space enables real-time correction without retraining or fine-tuning the pretrained model.
- Experiments on SeedTTS, LibriSpeech, and AIShell-1 demonstrate consistent improvements in textual alignment and perceptual quality, validating the effectiveness, efficiency, and practical generality of our method for robust zero-shot TTS generation.

2 Related Work

2.1 Zero-Shot Text-to-Speech

Zero-shot text-to-speech (TTS) aims to synthesize intelligible and natural speech for unseen speakers from short acoustic prompts, and recent progress has been largely driven by large-scale pre-training and data scaling in speech generation models [18, 28]. A representative line of work formulates zero-shot TTS as conditional generation from rich linguistic, semantic, and speaker-related representations, which has substantially improved speaker similarity, multilingual generalization, and overall naturalness [6, 11, 18, 30].

Recent systems also increasingly decouple high-level condition modeling from downstream acoustic generation, often combining language-model-style front ends with continuous speech generators to balance controllability and fidelity [11, 15]. In conditional generation more broadly, stronger conditioning representations have been shown to improve both fidelity and alignment, which further motivates studying the condition space itself rather than treating it as a fixed input throughout inference [24].

2.2 Flow Matching for Speech Generation

Flow Matching provides a simulation-free objective for continuous normalizing flows and has become an effective framework for high-fidelity generative modeling with efficient ODE-based sampling [20]. In speech synthesis, Matcha-TTS demonstrated that conditional flow matching can serve as a fast and high-quality non-autoregressive acoustic generator [21]. More recent TTS systems, including Voicebox, E2 TTS, F5-TTS, and ProsodyFlow, further showed that flow-based or flow-matching-based generation can achieve strong intelligibility and naturalness when paired with richer semantic or textual conditions [6, 13, 18, 30]. However, these systems still mainly follow an open-loop sampling paradigm in which conditional features are computed before generation and then reused throughout the trajectory, leaving limited room for online correction once the generation path begins to drift [6, 13, 18].

2.3 Inference-Time Guidance and Test-Time Optimization

A complementary research direction improves generative models at test time by introducing external guidance during sampling rather than modifying model parameters. Early work showed that classifier guidance can steer diffusion trajectories toward desired outputs through gradients from an auxiliary classifier [9], while subsequent studies demonstrated that classifier-free guidance (CFG) can achieve similar controllability by combining conditional and unconditional score estimates without requiring a separate classifier [17]. In TTS, recent studies have explored robustness improvement through reinforcement learning or preference optimization with external objectives, showing that intelligibility- and quality-related rewards can meaningfully improve synthesis behavior [25, 33]. At the same time, ASR systems and neural MOS predictors provide practical proxies for transcription accuracy and perceptual quality, making them suitable reward sources for closed-loop speech generation and evaluation [2, 14]. Different from post-training optimization or utterance-level reranking methods, our work focuses on gradient-based refinement of conditional features inside the sampling loop of a frozen flow-matching TTS model, enabling online correction without additional model training.

3 Preliminaries

3.1 Conditional Flow Matching for TTS

Recent zero-shot TTS systems often formulate acoustic generation as conditional transport from a simple prior distribution to the target acoustic distribution, followed by waveform reconstruction with a neural vocoder [6, 12, 13, 20]. Flow matching provides a continuous-time formulation for learning such transport through a velocity field [20]. Given a condition feature $\mathbf{c}$, let $\mathbf{x}_t$ denote the acoustic latent state at time $t \in [0, 1]$, where $\mathbf{x}_0 \sim p_0(\mathbf{x})$ is sampled from a simple prior and $\mathbf{x}_1$ denotes the terminal acoustic latent. In our implementation, this latent state is instantiated as a mel-space acoustic representation, which is decoded into waveform by the vocoder.

The conditional flow is governed by the probability-flow ODE

$$\frac{d\mathbf{x}_t}{dt} = v_\theta(\mathbf{x}_t, t, \mathbf{c}), \tag{1}$$

where v_θ is the velocity field parameterized by θ. Under a discrete solver with time steps $\{t_n\}_{n=0}^{N}$ such that $0 = t_0 < t_1 < \cdots < t_N = 1$, the sampling process is written as

$$\mathbf{x}_{t_n} = \mathbf{x}_{t_{n-1}} + \Delta t_n \, v_\theta(\mathbf{x}_{t_{n-1}}, t_{n-1}, \mathbf{c}), \qquad n = 1, \ldots, N, \tag{2}$$

where $\Delta t_n = t_n - t_{n-1}$. After solving Eq. (1), the terminal acoustic latent is converted into waveform by a vocoder,

$$\hat{\mathbf{w}} = \mathcal{V}(\mathbf{x}_1), \tag{3}$$

where $\mathcal{V}$ denotes the vocoder.

In standard conditional flow inference, the condition feature is computed once before sampling and then reused at every solver step [6, 12, 13, 30]. Our method departs from this fixed-condition setting and instead optimizes the condition online during sampling.

3.2 Reward Models

We consider two reward functions for inference-time guidance: an ASR reward for textual alignment [14] and a MOS reward for perceptual quality [2]. For notational convenience, let $r \in \{\text{asr}, \text{mos}\}$ denote the reward type, and define the unified reward as

$$R_r(\hat{\mathbf{w}}, y) := \begin{cases} R_{\text{asr}}(\hat{\mathbf{w}}, y), & r = \text{asr}, \\ R_{\text{mos}}(\hat{\mathbf{w}}), & r = \text{mos}, \end{cases} \tag{4}$$

where y is an inactive argument when $r = \text{mos}$.

4 Method

We propose Adaptive Condition Optimization for Text-to-Speech (Aco-TTS), an inference-time guidance framework for flow-based zero-shot TTS. Instead of keeping the condition feature fixed throughout sampling, we optimize it online with reward gradients defined on synthesized speech, while leaving model parameters unchanged.

4.1 Adaptive Condition Optimization

In this section, we formulate inference-time optimization of condition features during flow-based TTS sampling. Let E_ϕ denote a pretrained condition encoder with parameters ϕ. Given an input text y and a reference speech prompt p, the encoder produces a base condition feature

$$\mathbf{c}_{\text{base}} = E_\phi(y, p). \tag{5}$$

In standard inference, the condition remains fixed throughout generation [6, 12, 13, 30]. We instead allow the condition to vary with the flow time t, and denote the adaptive condition by $\mathbf{c}_t$.

Accordingly, the objective of Aco-TTS is expressed as

$$\max_{\mathbf{c}_{t_0}, \ldots, \mathbf{c}_{t_N}} \mathbb{E}_{\mathbf{x}_0 \sim p_0} \left[R_r(\hat{\mathbf{w}}, y) \right]. \tag{6}$$

Here, the expectation is taken over the initial latent $\mathbf{x}_0 \sim p_0$, and $\hat{\mathbf{w}}$ is obtained by solving Eq. (1) from $\mathbf{x}_0$ under the condition trajectory $\{\mathbf{c}_{t_n}\}_{n=0}^{N}$.

It should be noted that Eq. (6) optimizes only the final generated speech, so the intermediate latent states are not directly regularized. However, since $\mathbf{x}_1$ is the sequential sampling result of the entire flow trajectory, the optimization direction at each intermediate step can still be derived from the final reward objective. Because flow sampling proceeds sequentially from t_0 to t_N, the adaptive condition must also be determined sequentially. Hence, the practical sampling procedure turns Eq. (6) into a step-wise optimization problem.

Specifically, at sampling step t_n, we optimize the condition used at the current step based on the current latent state. We define the feasible set as

$$C_{t_n} := \left\{ \mathbf{c}_{t_n} : \|\mathbf{c}_{t_n} - \mathbf{c}_{\text{base}}\|_2 \le \rho \right\}, \tag{7}$$

where $\rho > 0$ is a scale hyperparameter that limits the update magnitude and keeps the optimized condition close to the original semantic anchor. Accordingly, the overall inference procedure can be viewed as a sequence of local optimization problems, where the condition at each step is recomputed from the base condition using the current latent state. Then the sequential optimization problem is

$$\max_{\mathbf{c}_{t_0} \in C_{t_0}} \cdots \max_{\mathbf{c}_{t_n} \in C_{t_n}} \cdots \max_{\mathbf{c}_{t_N} \in C_{t_N}} \mathbb{E}_{\mathbf{x}_0 \sim p_0} \left[R_r(\hat{\mathbf{w}}, y) \right]. \tag{8}$$

At step t_n, given the current latent state $\mathbf{x}_{t_{n-1}}$, the local objective can be written as

$$\max_{\mathbf{c}_{t_n} : \|\mathbf{c}_{t_n} - \mathbf{c}_{\text{base}}\|_2 \le \rho} \mathbb{E}_{\mathbf{x}_1 \sim p_\theta(\mathbf{x}_1 | \mathbf{x}_{t_{n-1}}, \mathbf{c}_{t_n})} \left[R_r(\mathcal{V}(\mathbf{x}_1), y) \right], \tag{9}$$

where $p_\theta(\mathbf{x}_1 \mid \mathbf{x}_{t_{n-1}}, \mathbf{c}_{t_n})$ denotes the terminal distribution induced by solving Eq. (1) from the current latent state $\mathbf{x}_{t_{n-1}}$ under condition $\mathbf{c}_{t_n}$. However, directly solving Eq. (9) is computationally expensive because each evaluation requires rolling out the flow trajectory from $\mathbf{x}_{t_{n-1}}$ to $\mathbf{x}_1$, and the reward must be evaluated on the resulting synthesized speech.

To address this, we apply a first-order Taylor approximation to the reward evaluated on the terminal latent. At step t_n, given the current latent state $\mathbf{x}_{t_{n-1}}$ and condition $\mathbf{c}_{t_n}$, we define the terminal mean latent, i.e., the mean estimate of the terminal latent $\mathbf{x}_1$ at step t_n, by

$$\bar{\mathbf{x}}_{1,t_n} := \mathbb{E}_{\mathbf{x}_1 \sim p_\theta(\mathbf{x}_1 | \mathbf{x}_{t_{n-1}}, \mathbf{c}_{t_n})} \left[\mathbf{x}_1 \right]. \tag{10}$$

Then, using a first-order Taylor expansion of $R_r(\mathcal{V}(\mathbf{x}_1), y)$ around $\bar{\mathbf{x}}_{1,t_n}$, we obtain

$$\begin{aligned} &\mathbb{E}_{\mathbf{x}_1 \sim p_\theta(\mathbf{x}_1 | \mathbf{x}_{t_{n-1}}, \mathbf{c}_{t_n})} \left[R_r(\mathcal{V}(\mathbf{x}_1), y) \right] \\ &\approx R_r\big(\mathcal{V}(\bar{\mathbf{x}}_{1,t_n}), y\big) + \mathbb{E}_{\mathbf{x}_1} \left[(\mathbf{x}_1 - \bar{\mathbf{x}}_{1,t_n})^\top \nabla_{\mathbf{x}} R_r\big(\mathcal{V}(\bar{\mathbf{x}}_{1,t_n}), y\big) \right] \\ &= R_r\big(\mathcal{V}(\bar{\mathbf{x}}_{1,t_n}), y\big), \end{aligned} \tag{11}$$

where the last equality holds because $\mathbb{E}_{\mathbf{x}_1}[\mathbf{x}_1 - \bar{\mathbf{x}}_{1,t_n}] = 0$. Therefore, we define the step-wise objective as

$$J_{t_n}^r(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{t_n}, y; \theta) := R_r\big(\mathcal{V}(\bar{\mathbf{x}}_{1,t_n}), y\big). \tag{12}$$

By construction, Eq. (12) reduces to the ASR-based objective when $r = \text{asr}$ and to the MOS-based objective when $r = \text{mos}$. For ease of presentation, we use $\bar{\mathbf{w}}_{t_n}$ as shorthand for the decoded waveform corresponding to $\mathcal{V}(\bar{\mathbf{x}}_{1,t_n})$.

Algorithm 1 Flow Matching with Aco-TTS

$\mathbf{x}_0 \sim p_0(\cdot)$
$\mathbf{c}_{\text{base}} \leftarrow E_\phi(y, p)$
for $n = 1$ to N **do**
  if $t_n \in$ {condition update steps} **then**
    $\hat{\mathbf{c}}_{t_n} \leftarrow \mathbf{c}_{\text{base}} + \rho \dfrac{\nabla_{\mathbf{c}} J_{t_n}^r(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{\text{base}}, y; \theta)}{\left\| \nabla_{\mathbf{c}} J_{t_n}^r(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{\text{base}}, y; \theta) \right\|_2}$
  else
    $\hat{\mathbf{c}}_{t_n} \leftarrow \mathbf{c}_{\text{base}}$
  end if
  $\mathbf{x}_{t_n} \leftarrow \mathbf{x}_{t_{n-1}} + \Delta t_n \, v_\theta(\mathbf{x}_{t_{n-1}}, t_{n-1}, \hat{\mathbf{c}}_{t_n})$
end for
return $\mathbf{x}_1$

4.2 Single-Step Condition Update

We present a single-step update of the adaptive condition at each sampling step to optimize Eq. (12) using the current latent state $\mathbf{x}_{t_{n-1}}$. Since evaluating this objective requires terminal prediction and reward backpropagation, the computational cost increases with each update. Therefore, we estimate the updated condition in a single gradient-based step.

We decompose the updated condition $\mathbf{c}_{t_n}$ into the base condition $\mathbf{c}_{\text{base}}$ and an update direction $\boldsymbol{\epsilon}_{t_n}$:

$$\mathbf{c}_{t_n} = \mathbf{c}_{\text{base}} + \boldsymbol{\epsilon}_{t_n}. \tag{13}$$

By expressing $\mathbf{c}_{t_n}$ in terms of $\boldsymbol{\epsilon}_{t_n}$, we reformulate the optimization problem in Eq. (12) into an equivalent problem over $\boldsymbol{\epsilon}_{t_n}$. Applying a first-order Taylor expansion around $\boldsymbol{\epsilon}_{t_n} = \mathbf{0}$ gives

$$\begin{aligned} \hat{\boldsymbol{\epsilon}}_{t_n} &:= \arg \max_{\substack{\boldsymbol{\epsilon}_{t_n} \\ \|\boldsymbol{\epsilon}_{t_n}\|_2 \le \rho}} J_{t_n}^r(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{\text{base}} + \boldsymbol{\epsilon}_{t_n}, y; \theta) \\ &\approx \arg \max_{\substack{\boldsymbol{\epsilon}_{t_n} \\ \|\boldsymbol{\epsilon}_{t_n}\|_2 \le \rho}} \boldsymbol{\epsilon}_{t_n}^\top \nabla_{\mathbf{c}} J_{t_n}^r(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{\text{base}}, y; \theta). \end{aligned} \tag{14}$$

The solution to Eq. (14) is given by

$$\begin{aligned} \hat{\boldsymbol{\epsilon}}_{t_n} &= \rho \frac{\nabla_{\mathbf{c}} J_{t_n}^r(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{\text{base}}, y; \theta)}{\left\| \nabla_{\mathbf{c}} J_{t_n}^r(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{\text{base}}, y; \theta) \right\|_2}, \\ \hat{\mathbf{c}}_{t_n} &= \mathbf{c}_{\text{base}} + \hat{\boldsymbol{\epsilon}}_{t_n}. \end{aligned} \tag{15}$$

That is, the updated condition is obtained by moving the base condition along the normalized reward gradient direction with step size ρ. After the condition update, the latent state is advanced according to

$$\mathbf{x}_{t_n} = \mathbf{x}_{t_{n-1}} + \Delta t_n \, v_\theta(\mathbf{x}_{t_{n-1}}, t_{n-1}, \hat{\mathbf{c}}_{t_n}). \tag{16}$$

This update refines the condition according to the current latent state and steers the generation trajectory toward higher textual alignment or better perceptual quality. In practice, the optimization can be applied at every sampling step, and it can also be restricted to selected steps to reduce computational overhead when needed. The overall inference procedure is summarized in Algorithm 1.

4.3 Theoretical Analysis

We provide a theoretical analysis of Aco-TTS from two perspectives: the optimality of condition optimization under the local objective,

and the influence of the condition update on the flow trajectory. Proofs and additional derivations are provided in the appendix.

We first show that the optimized condition is always no worse than the fixed base condition under the local objective in Eq. (9).

Proposition 1. Let

$$C_{t_n} = \left\{ \mathbf{c}_{t_n} : \|\mathbf{c}_{t_n} - \mathbf{c}_{\text{base}}\|_2 \le \rho \right\}. \tag{17}$$

Then,

$$\max_{\mathbf{c}_{t_n} \in C_{t_n}} J^r_{t_n}(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{t_n}, y; \theta) \ge J^r_{t_n}(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{\text{base}}, y; \theta). \tag{18}$$

Eq. (18) follows directly from the fact that $\mathbf{c}_{\text{base}} \in C_{t_n}$. Therefore, the optimal solution of the constrained local problem is guaranteed to achieve an objective value no lower than that of the fixed base condition. This shows that condition optimization is well motivated from the perspective of the local inference objective.

Next, we analyze the single-step update derived in Eq. (15). Recall that the updated condition is

$$\hat{\mathbf{c}}_{t_n} = \mathbf{c}_{\text{base}} + \rho\, \hat{\boldsymbol{\epsilon}}_{t_n}, \tag{19}$$

where

$$\hat{\boldsymbol{\epsilon}}_{t_n} := \frac{\nabla_{\mathbf{c}} J^r_{t_n}(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{\text{base}}, y; \theta)}{\left\| \nabla_{\mathbf{c}} J^r_{t_n}(\mathbf{x}_{t_{n-1}}, \mathbf{c}_{\text{base}}, y; \theta) \right\|_2}. \tag{20}$$

The following result shows that this update introduces a first-order guidance term into the flow dynamics.

Theorem 2. Assume that $v_\theta(\mathbf{x}, t, \mathbf{c})$ is continuously differentiable with respect to $\mathbf{c}$ in a neighborhood of $\mathbf{c}_{\text{base}}$. Then the velocity field under the updated condition $\hat{\mathbf{c}}_{t_n}$ can be expressed as

$$\begin{aligned} v_\theta(\mathbf{x}_{t_{n-1}}, t_{n-1}, \hat{\mathbf{c}}_{t_n}) &= v_\theta(\mathbf{x}_{t_{n-1}}, t_{n-1}, \mathbf{c}_{\text{base}}) \\ &\quad + \rho \frac{\partial v_\theta(\mathbf{x}_{t_{n-1}}, t_{n-1}, \mathbf{c}_{\text{base}})}{\partial \mathbf{c}}\, \hat{\boldsymbol{\epsilon}}_{t_n} + O(\rho^2). \end{aligned} \tag{21}$$

Eq. (21) follows from a first-order Taylor expansion of the velocity field around $\mathbf{c}_{\text{base}}$. It shows that the updated condition introduces a reward-dependent perturbation to the original flow dynamics. In other words, Aco-TTS not only modifies the condition representation itself, but also changes the instantaneous generation direction through the sensitivity of the flow model to the condition. When ρ is sufficiently small, this perturbation behaves as a first-order guidance term that steers the flow trajectory toward regions favored by the reward objective.

Finally, combining Eq. (16) with Eq. (21), the latent update can be written as

$$\begin{aligned} \mathbf{x}_{t_n} &= \mathbf{x}_{t_{n-1}} + \Delta t_n\, v_\theta(\mathbf{x}_{t_{n-1}}, t_{n-1}, \mathbf{c}_{\text{base}}) \\ &\quad + \Delta t_n\, \rho \frac{\partial v_\theta(\mathbf{x}_{t_{n-1}}, t_{n-1}, \mathbf{c}_{\text{base}})}{\partial \mathbf{c}}\, \hat{\boldsymbol{\epsilon}}_{t_n} + O(\rho^2). \end{aligned} \tag{22}$$

This expression makes clear that Aco-TTS can be interpreted as adding a reward-driven guidance term to the original flow trajectory. Therefore, the proposed condition optimization balances faithfulness to the pretrained flow dynamics with online correction toward better textual alignment or perceptual quality.

5 Experiments

5.1 Experimental Setup

Backbone and variants. We build Aco-TTS on the CosyVoice2 backbone [12]. CosyVoice2 is a multilingual zero-shot TTS system that combines large language model based condition modeling with a flow-based acoustic generator and neural vocoder decoding. We evaluate two variants of our method: Aco-TTS-ASR, which uses FunASR [14] as the ASR reward model to measure textual alignment, and Aco-TTS-MOS, which uses UTMOS-v2 [2] as the MOS reward model to measure perceptual quality. We also report the original CosyVoice2 system as the backbone baseline.

Datasets. We evaluate on four widely used zero-shot TTS test sets: SeedTTS test-en and test-zh from Seed-TTS [1], LibriSpeech test-clean using the TTS-oriented evaluation protocol in "evaluate-zero-shot-tts" [19], and AIShell-1 test [3]. These benchmarks cover both English and Chinese speech synthesis and provide a diverse test bed for evaluating intelligibility and perceptual quality. For AIShell-1, we do not use the full evaluation split; instead, we select a subset containing a few representative speakers in order to keep large-scale zero-shot evaluation tractable while preserving speaker diversity.

Baselines. We compare against recent strong zero-shot TTS systems, including MaskGCT [32], F5-TTS [6], Spark-TTS [31], IndexTTS2 [35], and CosyVoice2 [12]. Since CosyVoice2 is also the backbone of our method, the comparison between CosyVoice2 and Aco-TTS isolates the gain brought by inference-time condition optimization rather than architectural changes.

Metrics. We report Word Error Rate (WER) and Character Error Rate (CER) evaluated by FunASR [14] for textual alignment and automatic Mean Opinion Socre (MOS) evaluated by UTMOSv2 [2] for perceptual quality. We additionally report speaker similarity (SS), Gemini MOS, and human MOS. For SS, we use WavLM-based speaker verification for English [5] and ERes2NetV2-based speaker verification for Chinese [7]. Gemini MOS is evaluatied through Gemini [27], and Human MOS is evaluated with volunteers.

Implementation details. Unless otherwise specified, we use the full-step update setting and set the condition update step size to $\rho = 0.5$, according to the ablation in Fig. 4. All experiments are conducted on a Linux server with a single NVIDIA RTX 3090 GPU. Inference is implemented in PyTorch, and reward backpropagation is performed in full precision for numerical stability.

5.2 Comparison Results

Table 1 summarizes the comparison results across four zero-shot TTS benchmarks. Overall, Aco-TTS achieves a better balance between textual alignment and perceptual quality than the compared systems. Aco-TTS-ASR consistently improves WER/CER over the CosyVoice2 baseline on all four benchmarks. In particular, it achieves the lowest WER on LibriSpeech test-clean among all evaluated systems, demonstrating that ASR-based reward optimization is effective for improving textual alignment. Aco-TTS-MOS consistently improves MOS on all evaluated datasets, demonstrating that MOS-oriented reward optimization can improve perceptual quality without retraining the backbone model.

A notable observation is that IndexTTS2 achieves particularly strong CER on the Chinese benchmarks: SeedTTS test-zh and AIShell-1 test, where it surpasses our method. This may be partly due to stronger Chinese-specific optimization or data specialization in that system. However, its MOS and its English WER remain

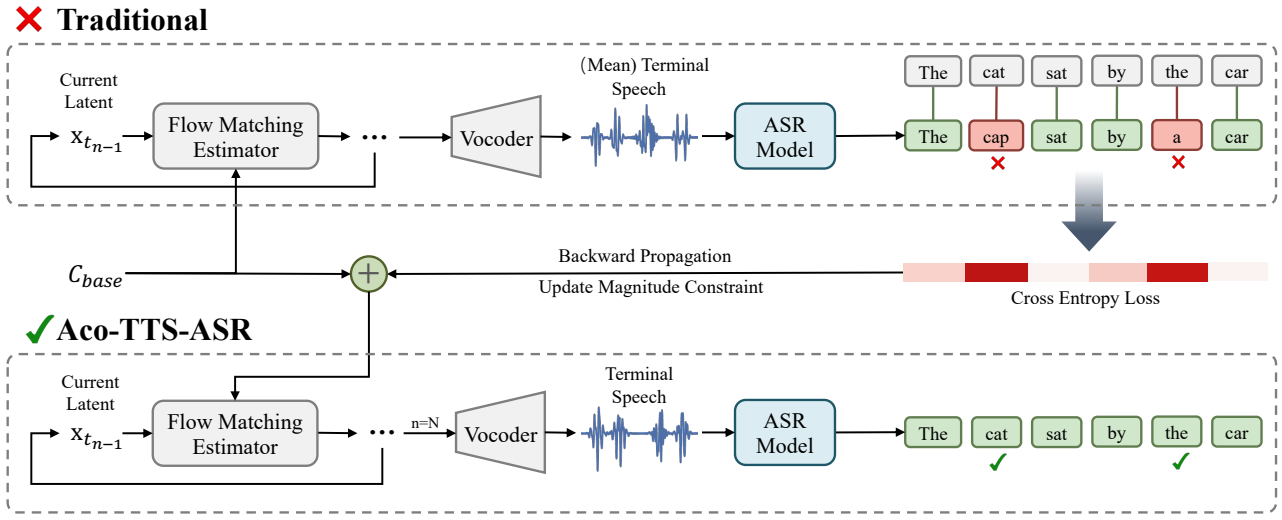

Figure 3: Illustration of the ASR-guided inference-time condition optimization. Compared to the traditional flow-matching pipeline (top) which may suffer from token-level misalignments or errors (e.g., "cap" vs. "cat"), our optimized approach (bottom) utilizes the differentiable ASR loss as feedback. Specifically, the cross-entropy loss from intermediate token mismatches is backpropagated to update the base condition c_{base}, guiding the subsequent ODE integration steps to produce more accurate and robust speech synthesis.

worse than those of Aco-TTS, suggesting that its advantages are less consistent across evaluation dimensions and languages.

F5-TTS is also competitive on English textual alignment and even outperforms our method on SeedTTS test-en WER. Nevertheless, its MOS is consistently lower than that of Aco-TTS, indicating that our method still provides a better overall trade-off between textual alignment and perceptual quality.

Speaker similarity scores are generally moderate across systems. This is partly because the CosyVoice2 baseline itself is not particularly strong on this dimension, and our method does not directly optimize speaker similarity. Importantly, the SS results do not show clear degradation after applying our reward-based inference-time optimization, suggesting that our method does not introduce obvious negative side effects on speaker consistency.

Finally, Gemini MOS and human MOS results further support the effectiveness of our method. Across all benchmarks, Aco-TTS-MOS consistently obtains competitive or superior perceptual ratings compared to the baseline systems, indicating that inference-time MOS reward optimization yields genuine improvements in perceptual quality rather than overfitting to the automatic MOS metric alone.

5.3 Ablation Studies

All ablation studies in this subsection are conducted on SeedTTS test-en using Aco-TTS-ASR, unless otherwise specified. Since these experiments optimize the ASR-based objective, we primarily focus on WER. MOS is reported only as an auxiliary metric to verify that the optimization does not introduce obvious reward hacking or severely degrade other aspects of speech quality.

Effect of step size ρ. We first study the effect of the condition update step size ρ on SeedTTS test-en under the ASR-based setting. As shown in Fig. 4, increasing ρ from 0 to 0.5 consistently reduces

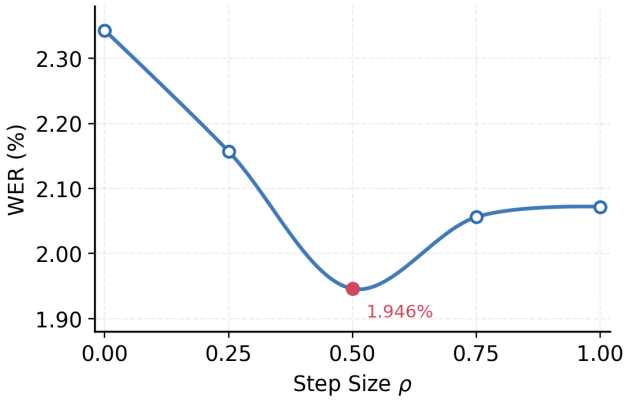

Figure 4: Effect of the condition update step size ρ on SeedTTS test-en with Aco-TTS-ASR. The best performance is achieved at $\rho = 0.5$.

WER, while larger values degrade performance. This trend indicates that moderate condition updates are beneficial, whereas overly large updates can push the condition away from a stable semantic region. Based on this result, we use $\rho = 0.5$ as the default setting in all subsequent experiments.

Effect of update steps. We further analyze which solver steps should receive reward-gradient updates. To study the trade-off between generation quality and efficiency, we compare three update schedules together with the baseline and the full-update setting. Specifically, *front-dense* updates Steps 1, 2, 3, 4, 6, 8, while *back-dense* updates Steps 2, 4, 7, 8, 9, 10; both use roughly 60% of all solver steps.

Table 1: Comparison results on zero-shot TTS benchmarks. We report WER for English datasets and CER for Chinese datasets. Aco-TTS-ASR denotes the ASR-reward variant, and Aco-TTS-MOS denotes the MOS-reward variant. Gemini MOS and human MOS are only reported for the three core systems: CosyVoice2, Aco-TTS-ASR, and Aco-TTS-MOS. The best result in each column is highlighted in bold, and the second-best result is highlighted with underlining.

Dataset	Model	WER/CER ↓	MOS ↑	SS ↑	Gemini MOS ↑	Human MOS ↑
SeedTTS test-en	MaskGCT	2.749%	2.918	**0.9537**	3.821±0.28	3.747±0.18
	F5-TTS	**1.886%**	3.093	0.9445	3.456±0.31	3.647±0.15
	SparkTTS	2.898%	3.145	0.9370	3.189±0.42	3.137±0.24
	IndexTTS2	2.250%	2.955	0.9530	3.712±0.24	3.727±0.16
	CosyVoice2	2.343%	3.327	0.8982	3.667±0.15	3.767±0.22
	Aco-TTS-ASR	1.946%	3.324	0.8957	3.733±0.13	3.792±0.20
	Aco-TTS-MOS	2.131%	**3.428**	0.8983	**3.767**±0.16	**3.933**±0.21
SeedTTS test-zh	MaskGCT	1.716%	2.760	0.9568	3.914±0.35	3.532±0.19
	F5-TTS	1.087%	2.994	**0.9574**	4.152±0.29	3.762±0.17
	SparkTTS	1.257%	3.048	0.9489	3.567±0.38	3.172±0.22
	IndexTTS2	**0.805%**	2.894	0.9562	4.288±0.21	3.922±0.14
	CosyVoice2	1.578%	3.163	0.7365	**4.300**±0.18	3.692±0.20
	Aco-TTS-ASR	1.046%	3.155	0.7373	3.900±0.12	3.933±0.16
	Aco-TTS-MOS	1.472%	**3.260**	**0.7387**	4.200±0.15	**3.967**±0.22
LibriSpeech test-clean	MaskGCT	4.206%	2.831	0.9376	3.412±0.33	4.087±0.15
	F5-TTS	1.933%	3.148	0.9031	3.658±0.26	4.017±0.18
	SparkTTS	5.037%	**3.508**	0.9289	3.324±0.41	4.047±0.19
	IndexTTS2	2.477%	3.234	**0.9418**	3.512±0.22	**4.187**±0.12
	CosyVoice2	1.761%	3.321	0.9079	3.533±0.14	4.067±0.16
	Aco-TTS-ASR	**1.457%**	3.319	0.9053	3.800±0.12	3.908±0.18
	Aco-TTS-MOS	1.838%	3.431	0.9062	**4.067**±0.10	3.983±0.13
AIShell-1 test	MaskGCT	2.990%	1.758	0.9141	3.156±0.36	2.687±0.12
	F5-TTS	2.694%	1.928	**0.9345**	3.528±0.24	3.267±0.21
	SparkTTS	1.536%	2.460	0.8987	2.942±0.45	2.807±0.18
	IndexTTS2	**0.964%**	2.182	0.9274	3.314±0.28	3.537±0.20
	CosyVoice2	2.064%	2.446	0.6718	3.467±0.19	3.417±0.25
	Aco-TTS-ASR	1.580%	2.440	0.6724	3.600±0.17	3.667±0.25
	Aco-TTS-MOS	2.294%	**2.622**	0.6707	**3.733**±0.13	**3.675**±0.19

As shown in Table 2, updating all steps achieves the best WER. Among the sparse schedules, the *back-dense* strategy performs better than the *front-dense* strategy. A plausible explanation is that the denoised result at very early steps is still too coarse to provide reliable reward guidance, whereas later steps produce clearer intermediate speech representations and therefore yield more meaningful gradient signals. This suggests that, when computational cost must be controlled, allocating updates to later solver steps is more effective than prioritizing earlier ones. Meanwhile, MOS remains broadly stable across all settings, indicating that the WER improvements are not obtained by noticeably harming perceptual quality.

Effect of reward formulation. We additionally compare two ASR-guided reward formulations. In the *Replace-Only* setting, gradients are backpropagated only through tokens whose predictions are incorrect, restricting the reward signal to misrecognized positions. In the *All Tokens* setting, the ASR loss is computed and backpropagated through all aligned tokens regardless of prediction correctness. Table 3 shows that *Replace-Only* yields only limited improvement over the baseline, whereas *All Tokens* brings a much larger WER reduction. This suggests that incorporating correctly predicted tokens into the reward signal is still beneficial, as it further strengthens pronunciation accuracy and stabilizes the generated text. In contrast,

Table 2: Ablation on update schedules on SeedTTS test-en with Aco-TTS-ASR. The best result is in bold, and the second-best result is underlined.

Update Schedule	WER ↓	MOS ↑
Baseline	2.3430%	3.3272
Back-Dense	2.0639%	3.3204
Front-Dense	2.3177%	**3.3317**
All Steps	**1.9460%**	3.3236

Table 3: Ablation on reward formulation on SeedTTS test-en with Aco-TTS-ASR. The best result is in bold, and the second-best result is underlined.

Reward Formulation	WER ↓	MOS ↑
Baseline	2.3430%	**3.3272**
Replace-Only	2.2420%	3.3251
All Tokens	**1.9460%**	3.3236

restricting the reward to incorrectly predicted tokens produces a much sparser optimization signal and is therefore less effective. MOS stays nearly unchanged across reward formulations, suggesting that the ASR-oriented gains do not come from obvious reward hacking.

5.4 Discussion

Overall, the experimental results demonstrate that Aco-TTS is effective as an inference-time optimization framework for flow-based zero-shot TTS. Without any additional training, the proposed method improves textual alignment or perceptual quality by dynamically refining the condition during sampling. The results further show that the choice of reward determines the direction of improvement, with Aco-TTS-ASR favoring intelligibility and Aco-TTS-MOS favoring perceptual quality.

6 Conclusion

We present Aco-TTS, a training-free inference-time framework that transforms the conventionally open-loop flow matching process into a closed-loop generation mechanism with online correction. Rather than fixing the condition feature prior to sampling, Aco-TTS dynamically refines it at each solver step via gradient signals derived from ASR and MOS reward models, steering the generation trajectory toward improved textual alignment or perceptual quality without modifying any pretrained model parameters.

Experiments on SeedTTS, LibriSpeech, and AIShell-1 consistently demonstrate reductions in word error rate and improvements in audio fidelity, with only modest additional modifications. Ablation studies further clarify the respective roles of step size, update scheduling, and reward formulation, offering practical guidance for deployment. Collectively, these results validate inference-time condition optimization as an effective and generalizable strategy for enhancing the robustness of zero-shot TTS systems, and suggest that gradient-based online refinement of the condition space represents a promising direction for future research in flow-based speech synthesis.

References

[1] Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. 2024. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430* (2024).

[2] Kaito Baba, Wataru Nakata, Yuki Saito, and Hiroshi Saruwatari. 2024. The t05 system for the voicemos challenge 2024: Transfer learning from deep image classifier to naturalness mos prediction of high-quality synthetic speech. In *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 818–824.

[3] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*. IEEE, 1–5.

[4] Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. 2023. The probability flow ode is provably fast. *Advances in Neural Information Processing Systems* 36 (2023), 68552–68575.

[5] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing* 16, 6 (2022), 1505–1518.

[6] Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. 2025. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 6255–6271.

[7] Yafeng Chen, Siqi Zheng, Hui Wang, Luyao Cheng, Qian Chen, Shiliang Zhang, and Junjie Li. 2024. ERes2NetV2: Boosting short-duration speaker verification performance with computational efficiency. *arXiv preprint arXiv:2406.02167* (2024).

[8] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. 2022. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11472–11481.

[9] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

[10] Chenpeng Du, Yiwei Guo, Hankun Wang, Yifan Yang, Zhikang Niu, Shuai Wang, Hui Zhang, Xie Chen, and Kai Yu. 2025. Vall-t: Decoder-only generative transducer for robust and decoding-controllable text-to-speech. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[11] Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. 2024. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407* (2024).

[12] Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. 2024. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117* (2024).

[13] Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al. 2024. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. In *2024 IEEE spoken language technology workshop (SLT)*. IEEE, 682–689.

[14] Zhifu Gao, Zerui Li, Jiaming Wang, Haoneng Luo, Xian Shi, Mengzhe Chen, Yabin Li, Lingyun Zuo, Zhihao Du, Zhangyu Xiao, et al. 2023. Funasr: A fundamental end-to-end speech recognition toolkit. *arXiv preprint arXiv:2305.11013* (2023).

[15] Wenhao Guan, Zhikang Niu, Ziyue Jiang, Kaidi Wang, Peijie Chen, Qingyang Hong, Lin Li, and Xie Chen. 2025. UniVoice: Unifying Autoregressive ASR and Flow-Matching based TTS with Large Language Models. *arXiv preprint arXiv:2510.04593* (2025).

[16] Changjin Han, Seokgi Lee, Gyuhyeon Nam, and Gyeongsu Chae. 2025. Improving Robustness of Diffusion-Based Zero-Shot Speech Synthesis via Stable Formant Generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[17] Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).

[18] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems* 36 (2023), 14005–14034.

[19] Keon Lee. 2024. *evaluate-zero-shot-tts*. doi:10.5281/zenodo.13340519

[20] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747* (2022).

[21] Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. 2024. Matcha-TTS: A fast TTS architecture with conditional flow matching. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 11341–11345.

[22] Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J Bryan. 2024. Ditto: Diffusion inference-time t-optimization for music generation. *arXiv preprint arXiv:2401.12179* (2024).

[23] Pinelopi Papalampidi, Olivia Wiles, Ira Ktena, Aleksandar Shtedritski, Emanuele Bugliarello, Ivana Kajic, Isabela Albuquerque, and Aida Nematzadeh. 2025. Dynamic classifier-free diffusion guidance via online feedback. *arXiv preprint arXiv:2509.16131* (2025).

[24] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* 35

(2022), 36479–36494.
[25] Xiaohui Sun, Ruitong Xiao, Jianye Mo, Bowen Wu, Qun Yu, and Baoxun Wang. 2025. F5r-tts: Improving flow-matching based text-to-speech with group relative policy optimization. *arXiv preprint arXiv:2504.02407* (2025).
[26] Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. 2024. Inference-time alignment of diffusion models with direct noise optimization. *arXiv preprint arXiv:2405.18881* (2024).
[27] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).
[28] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111* (2023).
[29] Hankun Wang, Chenpeng Du, Yiwei Guo, Shuai Wang, Xie Chen, and Kai Yu. 2024. Attention-constrained inference for robust decoder-only text-to-speech. In *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 630–637.
[30] Haoyu Wang, Sizhe Shan, Yinlin Guo, and Yuehai Wang. 2025. ProsodyFlow: High-fidelity Text-to-Speech through Conditional Flow Matching and Prosody Modeling with Large Speech Language Models. In *Proceedings of the 31st International Conference on Computational Linguistics*. 7748–7753.
[31] Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, et al. 2025. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *arXiv preprint arXiv:2503.01710* (2025).
[32] Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2024. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *arXiv preprint arXiv:2409.00750* (2024).
[33] Jixun Yao, Yuguang Yang, Yu Pan, Yuan Feng, Ziqian Ning, Jianhao Ye, Hongbin Zhou, and Lei Xie. 2025. Fine-grained preference optimization improves zero-shot text-to-speech. *arXiv preprint arXiv:2502.02950* (2025).
[34] Zhongqi Yue, Jiankun Wang, Qianru Sun, Lei Ji, Eric I Chang, Hanwang Zhang, et al. 2024. Exploring diffusion time-steps for unsupervised representation learning. *arXiv preprint arXiv:2401.11430* (2024).
[35] Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. 2026. Indextts2: A breakthrough in emotionally expressive and duration-controlled auto-regressive zero-shot text-to-speech. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 35139–35148.