DMOSpeech 2: Reinforcement Learning for Duration Prediction in Metric-Optimized Speech Synthesis

Yingahao Aaron Li[1]*, **Xilin Jiang**[1]*, **Fei Tao**[2], **Cheng Niu**[2],
Kaifeng Xu[2], **Juntong Song**[2], **Nima Mesgarani**[1]
[1]Columbia University, [2]NewsBreak

Abstract

Diffusion-based text-to-speech (TTS) systems have made remarkable progress in zero-shot speech synthesis, yet optimizing all components for perceptual metrics remains challenging. Prior work with DMOSpeech demonstrated direct metric optimization for speech generation components, but duration prediction remained unoptimized. This paper presents DMOSpeech 2, which extends metric optimization to the duration predictor through a reinforcement learning approach. The proposed system implements a novel duration policy framework using group relative preference optimization (GRPO) with speaker similarity and word error rate as reward signals. By optimizing this previously unoptimized component, DMOSpeech 2 creates a more complete metric-optimized synthesis pipeline. Additionally, this paper introduces teacher-guided sampling, a hybrid approach leveraging a teacher model for initial denoising steps before transitioning to the student model, significantly improving output diversity while maintaining efficiency. Comprehensive evaluations demonstrate superior performance across all metrics compared to previous systems, while reducing sampling steps by half without quality degradation. These advances represent a significant step toward speech synthesis systems with metric optimization across multiple components. The audio samples, code and pre-trained models are available at `https://dmospeech2.github.io/`.

1 Introduction

Text-to-speech (TTS) synthesis has progressed dramatically in recent years, with state-of-the-art systems producing speech virtually indistinguishable from human recordings [1, 2, 3]. Among the most significant advancements is zero-shot TTS, which is the ability to synthesize speech in the voice of an unseen speaker, given only a short audio sample without speaker-specific training. This capability has transformative potential across applications ranging from personalized digital assistants to accessibility tools and creative content production.

Despite impressive quality improvements, zero-shot TTS still faces a fundamental challenge: the lack of true end-to-end optimization for perceptual quality metrics. Current approaches struggle to directly optimize key metrics such as speaker similarity and intelligibility in an end-to-end manner, limiting their performance ceiling, especially for smaller and more efficient models. Reinforcement learning (RL) offers a potential indirect optimization approach [4, 5, 6, 7, 8] but comes with significant limitations. The ceiling of RL-based improvement is essentially best-of-N sampling [9], making its effectiveness heavily dependent on the original model's output diversity. For smaller, more efficient models with limited output diversity, RL may yield minimal improvements. Additionally, traditional RL for TTS imposes substantial computational overhead, as each training step requires generating complete speech samples—often through hundreds of sampling steps—making large-scale training prohibitively expensive without massive computational resources.

*These authors contributed equally. Correspondence: Y. A. Li (yl4579@columbia.edu).

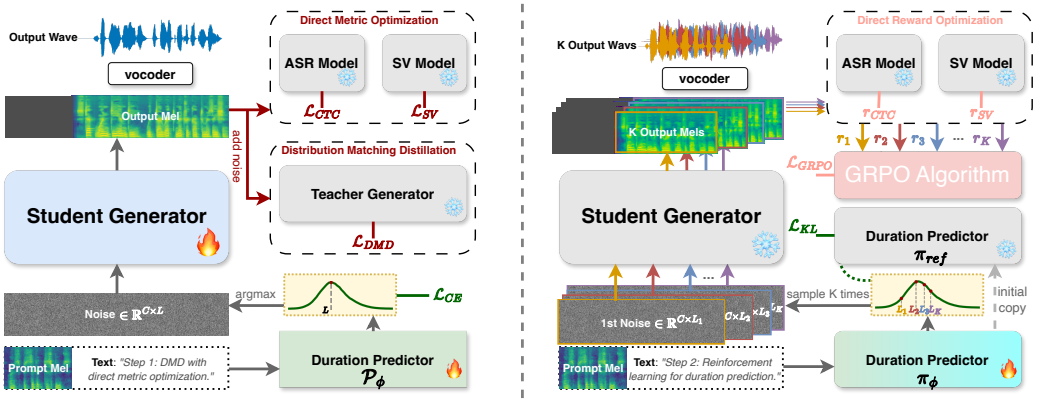

Figure 1: Overview of the DMOSpeech 2 framework. **(a) Left:** The original DMOSpeech architecture, where the duration predictor ($\mathcal{P}_\phi$) is trained self-supervisedly and separate from the TTS component, creating a disconnection that prevents end-to-end optimization. **(b) Right:** Our proposed DMOSpeech 2 framework, which employs Group Relative Policy Optimization (GRPO) to train the duration predictor with reinforcement learning (Algorithm 1), using speaker similarity and word error rate as reward signals, enabling end-to-end optimization of the entire TTS pipeline.

As the field has evolved, researchers have pursued two fundamentally different approaches to generating speech, each with their unique hurdles for direct metric optimization. Autoregressive models [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] generate speech step-by-step, similar to how large language models produce text. These systems naturally determine the duration of speech during generation but struggle with direct optimization due to the computational expense of backpropagating through their long generation sequences. While RL could theoretically help, these sequential models only amplify the previously mentioned limitations of RL approaches. Meanwhile, diffusion-based systems [21, 22, 23, 24, 25, 26, 27] take a different approach, treating speech synthesis as an inpainting task that requires knowing the total speech duration in advance. This creates a natural division in the pipeline: first predicting how long the speech should be, then generating the actual audio content. The challenge here is not just computational but also structural. Without a differentiable connection between these two components, traditional optimization techniques cannot flow through the entire system. Research has demonstrated that input durations significantly impact key metrics like speaker similarity (SIM) and word error rate (WER) [24], yet existing systems either train duration predictors separately from speech generation [21, 26] or use heuristic approaches based on prompt speaking rates [27, 24].

The original **D**irect **M**etric **O**ptimization **Speech** framework [28] made significant by enabling direct metric optimization for the speech generation component through diffusion model distillation. By reducing sampling steps from 128 to 4 and establishing direct gradient pathways within the generation process, DMOSpeech enabled direct optimization for speaker similarity and intelligibility. However, a critical limitation remained: the duration predictor component was still outside the optimization loop, creating a bottleneck in overall system quality.

This paper introduces **DMOSpeech 2**, which addresses the duration prediction challenge through reinforcement learning. We propose modeling the duration predictor as a probabilistic policy and applying reinforcement learning with group relative policy optimization (GRPO), using speaker similarity and word error rate as reward signals. Importantly, by applying RL specifically to the duration predictor and operating on samples generated by our efficient 4-step student model, we dramatically reduce the computational overhead typically associated with RL for TTS. This targeted approach also side-steps the limitations of whole-system RL, as optimizing duration prediction is a much more constrained problem than optimizing speech generation directly.

Additionally, to address the output diversity reduction observed in the original DMOSpeech as a consequence of distribution matching distillation [29], we introduce teacher-guided sampling, a hybrid approach that leverages the teacher model for initial denoising steps before transitioning to the student model. This strategy restores diversity to near-teacher levels while still achieving a 2$\times$ reduction in sampling steps and maintaining the significant quality improvements enabled by our direct metric optimization approach.

Using the flow-matching-based F5-TTS [27] as our teacher model, our comprehensive evaluations demonstrate that DMOSpeech 2 significantly outperforms both the previous system and other recent baselines across all metrics. The reinforcement learning approach to duration prediction results in particularly notable improvements in speaker similarity and word error rate, precisely targeting the limitations identified in previous systems.

The contributions of this work are twofold: 1) we propose a computationally efficient reinforcement learning framework specifically for duration prediction in non-parallel TTS systems, enabling alignment with perceptual metrics without the overhead typically associated with RL approaches, and 2) we propose a teacher-guided sampling for diffusion model distillation, restoring output diversity while maintaining computational efficiency. We will also make the source code and pre-trained models publicly available for future research in the community.

2 Related Works

Zero-Shot Text-to-Speech Synthesis Zero-shot TTS has evolved significantly over recent years, with approaches broadly categorized into two main paradigms. Early methods relied on speaker embeddings from pre-trained encoders [30, 31, 32, 33] or end-to-end speaker encoders [2, 34, 35, 36], but struggled with generalization due to their dependence on extensive feature engineering and with direct metric optimization due to their non-differentiable components such as duration predictors. Recent advancements have primarily focused on prompt-based approaches, which can be divided into autoregressive and diffusion-based methods. Autoregressive models [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] generate speech sequentially and naturally determine duration during generation, but face limitations in direct optimization due to the computational expense of backpropagation through long generation sequences. In contrast, diffusion-based approaches [21, 22, 23, 24, 25, 26, 27] treat speech synthesis as an inpainting task requiring predetermined speech duration, creating a natural division between duration prediction and actual speech generation. Although DMOSpeech [28] made progress by enabling direct optimization for the speech generation component, it still left the duration predictor outside the optimization loop. While duration inputs significantly impact metrics like speaker similarity and word error rate [24], existing systems either train duration predictors separately [21, 26] or use heuristic approaches based on prompt speaking rates [27, 24]. In DMOSpeech 2, we optimize the previously unoptimized duration predictor with reinforcement learning for perceptually relevant metrics.

Reinforcement Learning in Speech Synthesis Reinforcement learning (RL) has emerged as a promising approach for aligning speech synthesis systems with human perceptions, though its application to TTS presents unique challenges. Recent work has explored various RL techniques for improving TTS quality. SpeechAlign [5] introduced an iterative self-improvement strategy for neural codec language models that constructs preference datasets and optimizes toward human preferences. Similarly, UNO [4] proposed an uncertainty-aware optimization framework that integrates subjective human evaluation directly into the TTS training loop without requiring a separate reward model. Several approaches have focused on specific aspects of speech quality: [6] developed Emo-DPO for controllable emotional speech synthesis, differentiating subtle emotional nuances through preference optimization, while [7] demonstrated that direct preference optimization (DPO) consistently improves intelligibility and speaker similarity in LM-based TTS. Koel-TTS [8] enhanced encoder-decoder TTS models through preference alignment guided by automatic speech recognition and speaker verification. For diffusion-based TTS specifically, [37] introduced diffusion model loss-guided RL policy optimization (DLPO) to improve naturalness and quality, and [38] employed group relative policy optimization for flow-matching-based TTS models. However, these approaches incur substantial computational overhead, as each training step requires generating complete speech samples, often through hundreds of sampling steps, making large-scale training prohibitively expensive. Additionally, the effectiveness of RL is heavily dependent on the original model's output diversity, potentially yielding minimal improvements for smaller, more efficient models with limited diversity. Most existing approaches apply RL to the entire TTS pipeline, which exacerbates these challenges. DMOSpeech 2 addresses these limitations by specifically targeting RL to the duration predictor component, dramatically reducing computational overhead by operating on samples generated through an efficient 4-step student model, while simultaneously addressing the critical optimization gap in current non-parallel zero-shot TTS systems.

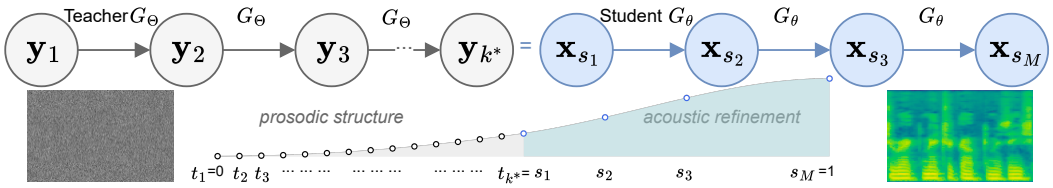

Figure 2: Illustration of teacher-guided sampling (Algorithm 2). The process begins with noise and uses the teacher model G_Θ for early denoising steps (gray circles) to establish prosodic structure up to a transition point t_{k^*}. Then, the student model G_θ (blue circles) takes over for the remaining steps to refine acoustic details in much fewer steps.

3 Methods

3.1 DMOSpeech with Flow Matching

DMOSpeech [28] is a framework for efficient zero-shot TTS that combines distribution matching distillation [29] with direct metric optimization. DMOSpeech 2 builds upon the original DMOSpeech framework while adopting F5-TTS [27] as the teacher model. This section summarizes the key components of our approach, highlighting the adaptations made for flow matching-based models. Fig. 1a illustrates the DMOSpeech architecture with details in Appendix B.

Unlike the original DMOSpeech which operated on latent representations from an audio autoencoder, DMOSpeech 2 directly generates mel-spectrograms, with waveforms synthesized using the pre-trained Vocos [39] vocoder. The framework consists of three training components. First, a student generator G_θ is trained through improved distribution matching distillation (DMD 2) [40] to match a pre-trained teacher model in distribution. This allows the student to generate high-quality speech with significantly fewer sampling steps (4 steps). Second, multi-modal adversarial training with a discriminator improves the perceptual quality of the generated speech. Finally, the direct metric optimization component enables end-to-end optimization of word error rate and speaker similarity metrics with pre-trained automatic speech recognition (ASR) models and speaker verification (SV) models on mel-spectrograms.

During inference, DMOSpeech generates speech directly from noise in four denoising steps, conditioned on the input text and speaker prompt and the total duration of the target speech. The process begins with sampling Gaussian noise $\mathbf{z} \sim \mathcal{N}(0, I)$ at a predefined duration L, which is determined by a separate duration predictor. The student generator G_θ then transforms this noise into mel-spectrograms through four sequential steps using the sway sampling schedule [27] with coefficient $u = -1$ at noise levels $t \in \{0.0000, 0.0761, 0.2929, 0.6173\}$ rather than uniform steps. The final spectrograms are converted to waveforms using the vocoder.

While DMOSpeech enabled direct metric optimization for the generator, it still maintained a critical limitation: the duration predictor remained outside the optimization loop. DMOSpeech 2 addresses this limitation through reinforcement learning, as detailed in the following sections.

3.2 Speech Length Predictor with RL

As established in the previous section, while DMOSpeech enables direct optimization of the speech generator, a critical limitation remains: the duration predictor sits outside the optimization loop, creating a disconnection that prevents end-to-end gradient-based optimization. This separation is particularly problematic because speech duration significantly impacts perceptual metrics like speaker similarity (SIM) and word error rate (WER) [24]. To address this limitation, DMOSpeech 2 introduces a novel reinforcement learning approach specifically targeting the speech length predictor.

3.2.1 Duration Predictor Architecture

We adopt an encoder-decoder transformer architecture similar to DiTTo-TTS [26] for our speech length predictor. Unlike conventional duration models that predict phoneme-level durations, our model is specifically designed to predict the total remaining length of speech to be generated.

Formally, let $\mathbf{x}$ represent the input text sequence and $\mathbf{p}_t$ represent the speech prompt up to frame t. Our speech length predictor $\mathcal{P}_\phi$ with parameters ϕ is trained to predict L_t, which is the number of remaining frames needed to complete the utterance:

$$P_\phi(L_t|\mathbf{x}, \mathbf{p}_t) = \mathcal{P}_\phi(\mathbf{x}, \mathbf{p}_t), \tag{1}$$

where L_t represents the length of the speech segment from frame t to the end. This formulation creates an autoregressive structure where the predicted remaining length decreases as the speech prompt extends. The architecture consists of a bidirectional text encoder that processes the input text to capture comprehensive contextual information. The decoder, equipped with causal masking to prevent future lookahead, takes the mel-spectrogram of the speech prompt as input. Cross-attention mechanisms integrate text features from the encoder, and the final layer applies softmax activation to predict a distribution over possible remaining lengths within a predefined maximum length. Our implementation uses a transformer with 4 encoder layers for text processing and 4 decoder layers with cross-attention mechanisms. The model employs 8 attention heads in each layer with a hidden dimension of 512. We set the maximum total duration to be 30 seconds binned into 300 possible duration classes, with increments of 100 ms.

During training, the ground truth label for the remaining audio length decreases by one at each subsequent time step. For a batch of sequences with mel-spectrogram lengths $\{L_1, L_2, ..., L_B\}$, where B is the batch size, the target remaining length is a decreasing sequence $(L_i-1, L_i-2, ..., 1, 0)$ for each training example L_i. The predictor is initially trained separately from the flow-matching model using cross-entropy loss between the predicted distribution and the ground truth remaining lengths. In DMOSpeech 2, we extend this training process with reinforcement learning to directly optimize for perceptual quality metrics.

3.2.2 GRPO-based Duration Optimization

To enable direct optimization for perceptual metrics, we formulate the speech length predictor as a stochastic policy in a reinforcement learning framework and apply group relative policy optimization (GRPO) [41], which allows us to optimize the length predictor directly for perceptual metrics without need of a differentiable pathway to the generator. The detailed algorithm is provided in Algorithm 1.

For each training instance $\mathbf{x}$ the input text and $\mathbf{p}$ the prompt, we define the policy for predicting the total speech length $\pi_\phi(L|\mathbf{x}, \mathbf{p}) = \mathcal{P}_\phi(\mathbf{x}, \mathbf{p})$. During training, we sample K different duration predictions for each input, where K is the group size:

$$L_k \sim \pi_\phi(L|\mathbf{x}, \mathbf{p}), \quad k = 1, 2, ..., K, \tag{2}$$

For each sampled duration, we generate speech using our efficient 4-step student model:

$$\mathbf{y}_k = G_\theta(\mathbf{z}, \mathbf{x}, \mathbf{p}, L_k), \quad z \sim \mathcal{N}(0, I), \tag{3}$$

where G_θ is our student generator and $\mathbf{z}$ is the initial noise. We then compute rewards for each generated speech sample using a combination of speaker similarity and speech recognition metrics:

$$r_k = \log p(\mathbf{x}|C(\mathbf{y}_k)) + \lambda_{\text{SIM}} \cdot \frac{\mathbf{e}_{\text{p}} \cdot \mathbf{e}_{y_k}}{\|\mathbf{e}_{\text{p}}\| \, \|\mathbf{e}_{y_k}\|}, \tag{4}$$

where $C(\cdot)$ is a pre-trained CTC-based ASR model operating on mel-spectrograms, $\mathbf{e}_{\text{p}} = S(\mathbf{p})$ and $\mathbf{e}_{y_k} = S(\mathbf{y}_k)$ are the speaker embeddings of the prompt and student-generated speech, and λ_{SIM} is the weighting factor. We chose $\lambda_{\text{SIM}} = 3$ to balance the contributions from the embedding similarity and word error rate (see Appendix A.2 for detailed discussion).

We normalize the reward to compute the advantage:

$$A_k = \frac{r_k - \mu_r}{\sigma_r}, \tag{5}$$

where μ_r and σ_r are the mean and standard deviation of rewards within the group.

In GRPO, we maintain three distinct policies. The current policy π_ϕ is the speech length predictor being actively trained. The old policy π_{old} is the version of the policy from which the current batch of samples was generated. In practice, this is typically the policy from several optimization steps ago. The reference policy π_{ref} is a frozen copy of the initially supervised model created at the beginning of

RL training and kept constant throughout the process to serve as an anchor for regularization. We define the ratio R_k as :

$$R_k = \frac{\pi_\phi(L_k|\mathbf{x}, \mathbf{p})}{\pi_{\text{old}}(L_k|\mathbf{x}, \mathbf{p})} \tag{6}$$

The GRPO loss for a single sample is:

$$\mathcal{L}_k = \min\left(A_k \cdot R_k, A_k \cdot \text{clip}\left(R_k, 1-\varepsilon, 1+\varepsilon\right)\right) - \beta \cdot \text{KL} \tag{7}$$

where $\varepsilon = 0.2$ is the clipping parameter that limits the policy update magnitude, $\beta = 0.04$ controls the strength of KL regularization, and $\text{KL} = \mathbb{D}_{\text{KL}}[\pi_\phi||\pi_{\text{ref}}]$ is the KL divergence between the current policy and the reference policy, preventing the trained policy from deviating too far from the initial model.

The full GRPO loss is defined as:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{\mathbf{x},\mathbf{p}}\left[\frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_k\right] \tag{8}$$

A challenge in applying RL to duration prediction is the potential for sparse rewards and limited exploration. If the model consistently predicts similar durations, it may fail to discover potentially superior alternatives. To address this, we incorporate temperature-based exploration during sampling. The Gumbel-softmax temperature parameter τ (set to 0.7 in our implementation) controls the entropy of the length distribution, with higher temperatures encouraging exploration of diverse length predictions:

$$\pi_\phi^\tau(L|\mathbf{x}, \mathbf{p}) = \frac{\exp(\log \pi_\phi(L|\mathbf{x}, \mathbf{p})/\tau)}{\sum_{L'} \exp(\log \pi_\phi(L'|\mathbf{x}, \mathbf{p})/\tau)} \tag{9}$$

We also implement a quality control mechanism that skips batches with insufficient reward diversity ($\max(r) - \min(r) < 0.01$), ensuring that the model only learns from batches where meaningful distinctions between good and bad duration predictions can be made. This approach prevents wasting computational resources on batches where all sampled durations yield similar quality speech, focusing training on examples where optimization can make a significant difference.

3.3 Teacher-Guided Sampling

3.3.1 Mode Shrinkage in Distribution Matching Distillation

One notable limitation of distribution matching distillation observed in the original DMOSpeech is a phenomenon we refer to as *mode shrinkage*. When student models are trained to generate speech in significantly fewer steps than their teacher, they tend to focus on high-probability regions of the data distribution, reducing diversity of the generated samples. While the student model exhibits similar mode coverage in sound quality compared to the teacher as indicated by the UTMOS [42] distributions, it demonstrates less diversity in prosodic features such as intonation patterns, rhythm variations, and speech cadences (Figure 3). This suggests that diversity reduction primarily occurs in the temporal and structural dimensions of speech rather than in its spectral characteristics.

The root cause of this diversity reduction can be traced to the diffusion process dynamics. In diffusion-based speech synthesis, different noise levels correspond to distinct aspects of the speech generation process. At high noise levels (early denoising steps), the model primarily establishes prosodic elements, phoneme durations, pauses, pitch contours, and text-speech alignments, essentially the semantic and structural framework of the utterance. In contrast, at low noise levels (later denoising steps), the model refines acoustic details such as voice quality, speaker identity, and spectral characteristics. When the student model is constrained to generate speech in just a few steps, it necessarily compresses this hierarchical generation process. Our empirical observations suggest that this compression disproportionately affects the diversity of prosodic and structural elements established in the early denoising phase.

Algorithm 1 GRPO-based Speech Length Predictor Training

Initialize speech length predictor $\mathcal{P}_\phi$ with supervised training
Create reference model π_{ref} as a frozen copy of initial model
Initialize batch queue $\mathcal{Q} \leftarrow []$
for step = 1 to max_steps **do**
  while size($\mathcal{Q}$) < 5 **do**
    Sample batch $(\mathbf{x}, \mathbf{p})$ from dataset
    Compute policy from model: $\pi_\phi \leftarrow \mathcal{P}_\phi(\mathbf{x}, \mathbf{p})$
    for $k = 1$ to K **do**
      $L_k \sim F_{\text{Gumbel}}(\pi_\phi, \tau)$
      $\mathbf{y}_k \leftarrow G_\theta(\mathbf{z}, \mathbf{x}, \mathbf{p}, L_k)$
      $r_k \leftarrow \log p(\mathbf{x}|C(\mathbf{y}_k)) + \lambda_{\text{SIM}} \cdot \frac{\mathbf{e}_\text{p} \cdot \mathbf{e}_{y_k}}{\|\mathbf{e}_\text{p}\| \|\mathbf{e}_{y_k}\|}$
    end for
    if $\max(r) - \min(r) > 0.01$ **then**
      for $k = 1$ to K **do**
        $A_k \leftarrow \frac{r_k - \mu_r}{\sigma_r}$
      end for
    else
      continue
    end if
    $\pi_{\text{old}} \leftarrow \pi_\phi$
    Push $([A_1, \ldots, A_K], [L_1, \ldots, L_K], \pi_{\text{old}})$ to $\mathcal{Q}$
  end while
  Dequeue $([A_1, \ldots, A_K], [L_1, \ldots, L_K], \pi_{\text{old}})$ from $\mathcal{Q}$
  $\pi_\phi \leftarrow \mathcal{P}_\phi(\mathbf{x}, \mathbf{p})$
  KL $\leftarrow \mathbb{D}_{\text{KL}}[\pi_\phi || \pi_{\text{ref}}]$
  Initialize loss $\mathcal{L} \leftarrow 0$
  for $k = 1$ to K **do**
    $R_k \leftarrow \frac{\pi_\phi(L_k|\mathbf{x},\mathbf{p})}{\pi_{\text{old}}(L_k|\mathbf{x},\mathbf{p})}$
    $R_{\text{clipped}} \leftarrow \text{clip}(R_k, 1-\varepsilon, 1+\varepsilon)$
    $\mathcal{L} \leftarrow \mathcal{L} - \frac{1}{K}(\min(A_k \cdot R_k, A_k \cdot R_{\text{clipped}}) - \beta \cdot \text{KL})$
  end for
  Update model parameters with gradient of $\mathcal{L}$
end for

3.3.2 Hybrid Sampling Strategy

To address the mode shrinkage problem, we introduce *teacher-guided sampling*, a hybrid approach that leverages the teacher model's diversity while preserving the student model's efficiency and improved speaker similarity from direct metric optimization. The core insight of our approach is to exploit the natural division of labor in the diffusion process: use the teacher model for early denoising steps on prosodic structure and the student model for acoustic refinement of later steps. Specifically, we employ the teacher model to perform the initial denoising steps up to a predefined noise level t_{switch}, which establihes diverse prosodic patterns and text-speech duration alignments. Then, we switch to the student model, which completes the remaining denoising process from t_{switch} to 1 in just a few efficient steps. This hybrid approach preserves the diversity benefits of the teacher model while still achieving significant computational savings.

Algorithm 2 outlines our teacher-guided sampling procedure. The process begins with random Gaussian noise $\mathbf{z}$ and progressively denoises it through a sequence of steps. The first K steps are performed by the teacher model using a flow matching formulation with the sway sampling schedule [27], which allocates more samples to early time steps where most of the semantic structure is established. Once the noise level reaches t_{switch}, the algorithm transitions to the student model, which completes the remaining denoising in just M steps (typically 2-3). A key advantage of our approach is that it achieves a more favorable trade-off between computational efficiency and output diversity. By delegating the labor-intensive task of establishing prosodic structure to the teacher model and the refinement of acoustic details to the student model, we leverage the strengths of both approaches.

Algorithm 2 Teacher-Guided Sampling

Require: Teacher model G_Θ, student model G_θ, teacher steps K, student steps M, switching time t_{switch}, text embedding $\mathbf{x}$, prompt embedding $\mathbf{p}$, duration L, CFG strength λ

Sample $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ with length L
Initialize $\mathbf{y}_0 \leftarrow \mathbf{z}$
Generate teacher time steps $\{t_1, t_2, \ldots, t_K\}$ using sway sampling, $t_1 = 0$
Find index k^* such that $t_{k^*} \leq t_{\text{switch}} < t_{k^*+1}$
for $k = 1$ to k^* **do**
    $\mathbf{v}_k \leftarrow G_\Theta(\mathbf{y}_{k-1}, \mathbf{x}, \mathbf{p}, t_k)$
    $\mathbf{y}_k \leftarrow \mathbf{y}_{k-1} + (t_k - t_{k-1}) \cdot \mathbf{v}_k$
end for
Generate student time steps $\{s_1, s_2, \ldots, s_M\}$ with $s_1 = t_{k^*}$ and $s_M = 1$
$\mathbf{x}_{s_1} \leftarrow \mathbf{y}_{k^*}$
for $m = 1$ to M **do**
    $\hat{\mathbf{x}}_1^m \leftarrow G_\theta(\mathbf{x}_{s_m}; \mathbf{x}, \mathbf{p}, s_m)$
    if $m < M$ **then**
        Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
        $\mathbf{x}_{s_{m+1}} \leftarrow (1 - s_{m+1})\boldsymbol{\epsilon} + s_{m+1}\hat{\mathbf{x}}_1^m$
    end if
end for
$\hat{\mathbf{x}}_1^M \leftarrow G_\theta(\mathbf{x}_{s_M}; \mathbf{x}, \mathbf{p}, s_M)$
return $\hat{\mathbf{x}}_1^M$

The teacher model is employed for fewer steps than its typical full inference (approximately 6-14 steps instead of 32), while the student model still performs only a small number of denoising steps (2-3 instead of 4).

Our empirical evaluation (Table 1) confirms that teacher-guided sampling successfully mitigates the mode shrinkage problem, restoring the diversity of the generated speech to levels comparable to the teacher model, particularly in terms of pitch variation and cadence diversity. Notably, this improvement comes with only a modest increase in computational cost compared to the pure student model but still $1.8\times$ faster than the full teacher model. Additionally, similar to the student model, our hybrid approach produces samples with better SIM and WER than the teacher-only samples, benefiting from the direct metric optimization of the DMOSpeech framework.

The parameters K, t_{switch}, and M offer flexible control over the trade-off between computational efficiency and output diversity. For applications where diversity is critical, such as creative content production, a higher t_{switch} value (around 0.4-0.5) can be used, allocating more steps to the teacher model. Conversely, for applications where efficiency is paramount, such as real-time systems, a lower t_{switch} value (around 0.1-0.2) can be employed with minimal degradation in perceptual quality.

4 Experiments

4.1 Experimental Setup

Datasets Following F5-TTS [27], we utilize the in-the-wild multilingual speech dataset Emilia [43] to train our models. After filtering out transcription failures and misclassified language speech, we retain approximately 95k hours of English and Chinese data. For evaluation, we adopt three test sets: Seed-TTS [44] *test-en* with 1088 samples from CommonVoice [45], and Seed-TTS *test-zh* with 2020 samples from DiDiSpeech[46].

Training For our teacher model, we adopt F5-TTS [27] with approximately 300M parameters, trained for 2M steps on the Emilia dataset. We maintain the same hyperparameter configuration as in the original F5-TTS, with a batch size of 307,200 audio frames (0.91 hours), using the AdamW optimizer [47] with a peak learning rate of 7.5e-5, linear warmup for 20K updates, and linear decay afterwards. For the student model training in DMOSpeech 2, we follow the approach in [28] but use half the batch size of the teacher model training. The learning rate for the student model resumes from the final learning rate of the teacher model training (around 6e-5) and continues for an additional

Table 1: Objective and subjective evaluation results on *Seed-TTS-en* and *Seed-TTS-zh* evaluation sets. CMOS-S and CMOS-N refer to CMOS for similarity and naturalness, respectively, with DMOSpeech 2 (our system with 4 sampling steps) as the anchor (negative means DMOSpeech 2 is better). The **best** values for objective evaluations are shown in **bold** and the second-best values are underlined where S/A stands for the same as above. For subjective evaluations, the statistically significant results are marked by one asterisk if $p < 0.05$ and two asterisks if $p < 0.01$. CV_{f_0} is computed with the DDPM sampler for fairness.

Model	*Seed-TTS-en*		*Seed-TTS-zh*		English		Chinese		CV_{f_0} ↑	RTF↓
	WER↓	SIM↑	CER↓	SIM↑	CMOS-N	CMOS-S	CMOS-N	CMOS-S		
Ground Truth	2.143	0.734	1.254	0.755	0.03	-0.13^{*}	0.02	-0.06	—	—
F5-TTS Teacher (32 steps)	1.947	0.662	1.695	0.750	-0.12^{*}	-0.04	-0.09	-0.11^{*}	**0.6659**	0.1671
DMOSpeech 2 (4 steps)	1.752	0.698	1.527	**0.760**	0.0	0.0	0.0	0.0	0.4640	**0.0316**
w/o duration predictor RL	3.750	0.672	2.000	0.750	-0.43^{**}	-0.48^{**}	-0.26^{*}	-0.31^{*}	S/A	S/A
Teacher-Guided (16 steps)	**1.738**	**0.699**	**1.468**	**0.760**	0.01	-0.03	0.45^{**}	0.3^{*}	0.5932	0.0941

200K steps on the Emilia dataset. The duration predictor uses an encoder-decoder transformer architecture similar to DiTTo-TTS [26]. It is initially trained on the Emilia dataset for 85K steps with a learning rate of 1e-4 and the same batch size as the F5-TTS teacher training. We use the AdamW optimizer with default parameters of Pytorch. After this initial training, we further fine-tune the duration predictor using GRPO [38] for an additional 1.5K steps with a group size of 16, optimizing directly for speaker similarity and word error rate metrics. All experiments were conducted on 8 NVIDIA H100 GPUs.

Baselines We compare several configurations of our models with both subjective and objective evaluations: (1) The ground truth recordings, (2) F5-TTS teacher without a duration predictor using 32 sampling steps, (3) DMOSpeech 2 with the RL-optimized duration predictor using 4 sampling steps, (4) student with the duration predictor before RL using 4 sampling steps, and (5) a teacher-guided sampling approach where the teacher model handles initial denoising steps before transitioning to the student model ($t_{switch} = 0.25$, with teacher handling 14 steps and student handling 2 steps, for a total of 16 steps). We use the pretrained Vocos vocoder [39] to convert generated mel-spectrograms to audio signals. We also compare our DMOSpeech 2 with several state-of-the-art TTS systems on objecetive metrics: CosyVoice 2 [16], Spark-TTS [18], LLaSA-8B [20], MaskGCT [13], and our F5-TTS teacher model (32 steps) [27]. All samples were resampled to 24 kHz for a fair comparison.

4.2 Evaluation Metrics

We evaluate our models under the *cross-sentence* task, following the protocol established in [21]. In this task, the model is given a reference text, a short speech prompt, and its transcription, and is required to synthesize speech reading the reference text while mimicking the voice characteristics of the prompt speaker.

For objective evaluation, we report word error rate (WER) and speaker similarity between generated and original target speeches (SIM). For WER, we employ Whisper-large-v3 [48] to transcribe English and Paraformer-zh [49] for Chinese, following the approach in Seed-TTS [44]. For SIM-o, we use a WavLM-large-based [50] speaker verification model to extract speaker embeddings for calculating the cosine similarity between synthesized and ground truth speeches. We also measure the real-time factor (RTF) to evaluate inference speed, defined as the ratio of speech generation time to the duration of the generated speech on a single H100 GPU. Additionally, to demonstrate that teacher-guided sampling helps improve sampling diversity, we compare the coefficient of variation of the pitch (CV_{f_0}) of 50 different samples synthesized with the same input text and prompt across 20 text-prompt pairs for various configurations of our models averaged across all frames (with the same input total duration). For the teacher, we used DDPM [51] modified for flow-matching [52] to have a fair comparison with the students as they have additional noise injections throughout the sampling process (see Algorithm 3 for more details).

For subjective evaluation, we conduct human listening tests using comparative mean opinion scores (CMOS) for both naturalness and similarity. For CMOS, human evaluators are presented with randomly ordered synthesized speech from one model and an anchor model (our DMOSpeech 2 with the RL-optimized duration predictor using 4 sampling steps), and are asked to rate which sample has

Table 2: Comparison with state-of-the-art models on *Seed-TTS-en* and *Seed-TTS-zh* evaluation sets. The **best** values in each column are shown in **bold** and the second-best values are underlined. All samples from baseline models were synthesized using the official checkpoints released by the authors.

Model	#Params	Dataset (# Hours)	*Seed-TTS-en*		*Seed-TTS-zh*		RTF↓
			WER↓	SIM↑	CER↓	SIM↑	
Ground Truth	–	–	2.143	0.734	1.254	0.755	–
F5-TTS (32 steps) [27]	0.3B	Emilia [43] (95k hrs)	1.947	0.662	1.695	0.750	0.167
CosyVoice 2 [16]	0.5B	Proprietary (200k hrs)	3.358	0.641	1.582	0.754	0.527
Spark-TTS [18]	0.5B	VoxBox [18] (100k hrs)	2.308	0.572	1.717	0.657	1.784
MaskGCT [13]	0.7B	Emilia [43] (95k hrs)	2.622	**0.713**	2.395	**0.772**	2.397
LLaSA-8B [20]	8B	Proprietary (200k hrs)	3.994	0.594	4.214	0.671	1.374
DMOSpeech 2 (Student-Only, 4 steps)	0.3B	Emilia [43] (95k hrs)	1.752	0.698	1.527	0.760	**0.032**
DMOSpeech 2 (Teacher-Guided, 16 steps)	0.6B	Emilia [43] (95k hrs)	**1.738**	0.699	**1.468**	0.760	0.094

higher similarity with respect to the prompt speech and more like a human recording (either $+1$ or -1). We report the average scores of a total of 320 samples in both English and Chinese. For more details, we refer the readers to Appendix C.

4.3 Results

4.3.1 Main Results

Table 1 shows that DMOSpeech 2 with the RL-optimized duration predictor significantly outperforms both the teacher model and the student model without duration predictor optimization. On the English evaluation set, DMOSpeech 2 achieves a WER of 1.752 compared to 1.947 for F5-TTS and 3.750 for DMOSpeech without RL optimization. For speaker similarity, DMOSpeech 2 reaches 0.698 compared to 0.662 for F5-TTS (teacher) and 0.672 for DMOSpeech without RL. We observe similar improvements on the Chinese evaluation set, where DMOSpeech 2 achieves a CER of 1.527 and similarity of 0.760, outperforming both F5-TTS with 1.695 CER and 0.750 SIM, and DMOSpeech without RL with 2.000 CER and 0.750 SIM.

Most remarkably, DMOSpeech 2 delivers this superior performance while maintaining exceptional computational efficiency, with an RTF of 0.0316, which is more than $5\times$ faster than the teacher model's 0.1671. The teacher-guided sampling approach achieves slightly better objective metrics with WER of 1.738 and CER of 1.468 but an increased computation time.

The subjective CMOS evaluation further confirms our approach's effectiveness. Human evaluators rated DMOSpeech 2 significantly better than that without RL (i.e., the original DMOSpeech), with substantial margins in both English and Chinese. For English, DMOSpeech 2 showed naturalness superiority with CMOS-N of -0.43 and similarity advantage with CMOS-S of -0.48, both statistically significant at $p < 0.01$. For Chinese, we observed similar benefits with CMOS-N of -0.26 and CMOS-S of -0.31, significant at $p < 0.05$. DMOSpeech 2 also outperforms F5-TTS, achieving significantly better English naturalness with CMOS-N of -0.12 and Chinese similarity with CMOS-S of -0.11, both at $p < 0.05$. Interestingly, while the teacher-guided sampling approach shows comparable performance to DMOSpeech 2 for English, it demonstrates significantly better subjective scores for Chinese, with CMOS-N reaching $+0.45$ at $p < 0.01$ and CMOS-S of $+0.3$ at $p < 0.05$. Perhaps most importantly, DMOSpeech 2 achieves results statistically indistinguishable from ground truth recordings in naturalness for both English and Chinese. For English similarity, it even achieves a noteworthy CMOS-S of -0.13 compared to ground truth, significant at $p < 0.05$. These results confirm that our approach produces speech that approaches human-level quality on the evaluation benchmark dataset while maintaining exceptional computational efficiency.

4.3.2 Comparison with State-of-the-Art Models

Table 2 show the comparison of DMOSpeech 2 with previous state-of-the-art TTS models on the Seed-TTS evaluation sets. DMOSpeech 2, in both its student-only and teacher-guided variants, significantly outperforms most baseline models in terms of intelligibility while maintaining competitive speaker similarity and vastly superior computational efficiency. Our student-only DMOSpeech 2 model achieves an English WER of 1.752 and a Chinese CER of 1.527, substantially better than all baseline models with similar or larger parameter counts. The next best performer, our teacher model F5-TTS,

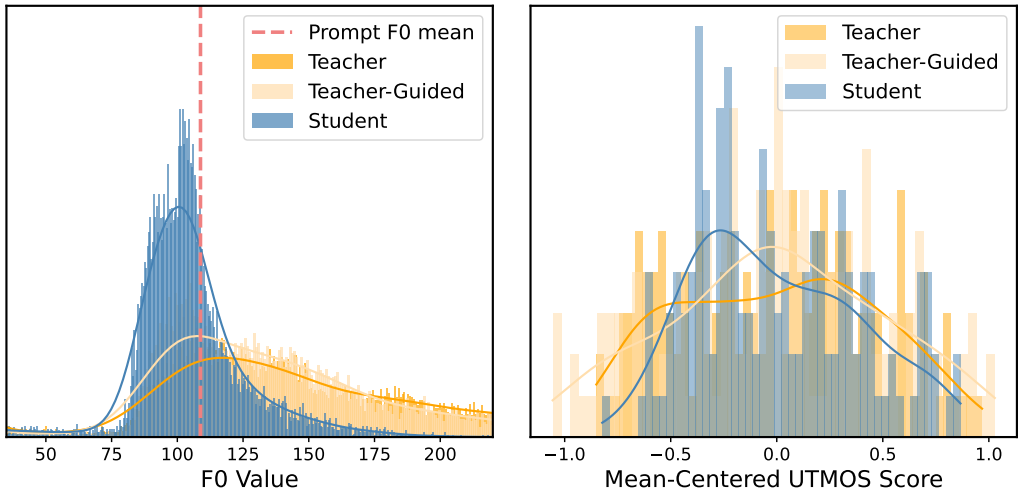

Figure 3: Comparison of diversity across sampling methods. (a) F0 value distributions (shown as histograms and kernel density estimates). The student model (light blue) exhibits a much narrower distribution compared to the teacher model (yellow), indicating mode shrinkage in prosodic patterns. The teacher-guided approach (orange) successfully recovers much of this diversity. (b) Mean-centered UTMOS score distributions. Acoustic quality remains consistent across all models despite differences in prosodic diversity, supporting our hypothesis that diversity reduction primarily affects prosodic and temporal aspects rather than acoustics.

achieves a WER of 1.947 and CER of 1.695 with the same parameter count but requires $5.3\times$ more computation time. The teacher-guided variant further improves these results to 1.738 WER and 1.468 CER while still maintaining a $1.8\times$ speed advantage over the teacher model F5-TTS despite requiring twice the parameter size (from 0.3B to 0.6B), as it needs the weight of both the teacher and the student models. In terms of speaker similarity, DMOSpeech 2 variants score 0.698-0.699 for English and 0.760 for Chinese, outperforming most baselines except MaskGCT, which achieves the highest similarity scores but at the cost of significantly worse intelligibility and dramatically higher computational requirements. MaskGCT has an RTF of 2.397, making it $75\times$ slower than DMOSpeech 2.

It is noteworthy that DMOSpeech 2 outperforms much larger models like LLaSA-8B across all metrics, despite having only 0.3B parameters compared to 8B. This demonstrates that our targeted optimization approach through reinforcement learning of the duration predictor is more effective than simply scaling up model size. The computational efficiency of DMOSpeech 2 is particularly striking, with an RTF of 0.032 for the student-only variant, making it $5.2\times$ faster than F5-TTS, $16.5\times$ faster than CosyVoice 2, $55.8\times$ faster than Spark-TTS, and $42.9\times$ faster than LLaSA-8B. This exceptional efficiency makes DMOSpeech 2 particularly suitable for real-time applications and deployment on resource-constrained devices.

4.3.3 Effect of Teacher-Guided Sampling on Diversity

As shown in Table 1, teacher-guided sampling successfully addresses diversity limitations in our distilled student model. The coefficient of variation of pitch (CV_{f_0}) reveals the teacher model's superior diversity (0.6659) compared to the student model's reduced variation (0.4640, a 30.3% decrease), indicating the student model suffers from mode shrinkage. Our teacher-guided approach recovers much of this diversity (0.5932, 89.1% of teacher's diversity) while maintaining superior WER and speaker similarity from the student model with direct metric optimization. Figure 3a illustrates this effect through F0 distributions. The student model shows a narrower, more peaked distribution than the teacher model, demonstrating mode shrinkage from aggressive step reduction. The teacher-guided approach successfully broadens this distribution. In Figure 3b, we plot the mean-centered UTMOS score distributions since different models demonstrate significant differences in their mean UTMOS scores. Despite this, the mean-centered distributions after remain consistent across all models, indicating diversity reduction occurs primarily in prosodic aspects rather than

spectral characteristics. This hybrid approach achieves a favorable trade-off between computational efficiency (RTF = 0.0941) and output diversity by leveraging the teacher model for establishing prosodic structure and the student model for efficient acoustic refinement.

5 Conclusion

This paper introduces DMOSpeech 2, which addresses two critical limitations in end-to-end diffusion-based TTS systems: optimizing the duration predictor component for perceptual metrics and mitigating diversity reduction in distilled models. Through reinforcement learning with GRPO, we optimize the duration predictor directly for speaker similarity and intelligibility, while our teacher-guided sampling approach restores prosodic diversity. Comprehensive evaluations show that DMOSpeech 2 significantly outperforms previous state-of-the-art models across various metrics while maintaining exceptional computational efficiency. The ability to optimize the previously isolated duration predictor component marks significant progress in end-to-end TTS optimization. Future work could explore applying our targeted RL approach to other components in generative pipelines that are difficult to optimize directly with gradient descent, such as the teacher model in the hybrid sampling approach, and employing rewards other than WER and SIM to align our models with human perceptions further.

DMOSpeech 2 raises important societal considerations. Our system's improved speaker similarity and intelligibility offer significant benefits for accessibility, personalized assistants, and content creation. However, like all high-fidelity voice synthesis technologies, it presents potential risks for voice spoofing and deepfakes. The computational efficiency of our approach also democratizes access to this technology, amplifying both benefits and risks. To address these concerns, we emphasize the importance of developing more robust detection methods for synthetic speech and establishing appropriate governance frameworks. To foster further research and reproducibility, we will release our source code and pre-trained models publicly. We believe our open-source approach will accelerate progress in addressing both the technical challenges and ethical considerations associated with advanced TTS systems.

References

[1] Xu Tan, Jiawei Chen, Haohe Liu, Jian Cong, Chen Zhang, Yanqing Liu, Xi Wang, Yichong Leng, Yuanhao Yi, Lei He, et al. Naturalspeech: End-to-end text-to-speech synthesis with human-level quality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[2] Yinghao Aaron Li, Cong Han, Vinay Raghavan, Gavin Mischler, and Nima Mesgarani. Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, et al. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. *arXiv preprint arXiv:2403.03100*, 2024.

[4] Chen Chen, Yuchen Hu, Wen Wu, Helin Wang, Eng Siong Chng, and Chao Zhang. Enhancing zero-shot text-to-speech synthesis with human feedback. *arXiv preprint arXiv:2406.00654*, 2024.

[5] Dong Zhang, Zhaowei Li, Shimin Li, Xin Zhang, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechalign: Aligning speech generation to human preferences. *arXiv preprint arXiv:2404.05600*, 2024.

[6] Xiaoxue Gao, Chen Zhang, Yiming Chen, Huayun Zhang, and Nancy F Chen. Emo-dpo: Controllable emotional speech synthesis through direct preference optimization. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[7] Jinchuan Tian, Chunlei Zhang, Jiatong Shi, Hao Zhang, Jianwei Yu, Shinji Watanabe, and Dong Yu. Preference alignment improves language model-based tts. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[8] Shehzeen Hussain, Paarth Neekhara, Xuesong Yang, Edresson Casanova, Subhankar Ghosh, Mikyas T Desta, Roy Fejgin, Rafael Valle, and Jason Li. Koel-tts: Enhancing llm based speech generation with preference alignment and classifier free guidance. *arXiv preprint arXiv:2502.05236*, 2025.

[9] Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, Kenshi Abe, Mitsuki Sakamoto, and Eiji Uchibe. Evaluation of best-of-n sampling strategies for language model alignment. *arXiv preprint arXiv:2502.12668*, 2025.

[10] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

[11] Puyuan Peng, Po-Yao Huang, Daniel Li, Abdelrahman Mohamed, and David Harwath. Voicecraft: Zero-shot speech editing and text-to-speech in the wild. *arXiv preprint arXiv:2403.16973*, 2024.

[12] Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *arXiv preprint arXiv:2406.05370*, 2024.

[13] Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *arXiv preprint arXiv:2409.00750*, 2024.

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

[15] Chenpeng Du, Yiwei Guo, Hankun Wang, Yifan Yang, Zhikang Niu, Shuai Wang, Hui Zhang, Xie Chen, and Kai Yu. 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)*, pages 1–5. IEEE, 2025.

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

[17] Xinfa Zhu, Wenjie Tian, and Lei Xie. Autoregressive speech synthesis with next-distribution prediction. *arXiv preprint arXiv:2412.16846*, 2024.

[18] Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, et al. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *arXiv preprint arXiv:2503.01710*, 2025.

[19] Xingchen Song, Mengtao Xing, Changwei Ma, Shengqiang Li, Di Wu, Binbin Zhang, Fuping Pan, Dinghao Zhou, Yuekai Zhang, Shun Lei, et al. Touchtts: An embarrassingly simple tts framework that everyone can touch. *arXiv preprint arXiv:2412.08237*, 2024.

[20] Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi DAI, et al. Llasa: Scaling train-time and inference-time compute for llama-based speech synthesis. *arXiv preprint arXiv:2502.04128*, 2025.

[21] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36, 2024.

[22] Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv preprint arXiv:2304.09116*, 2023.

[23] Yinghao Aaron Li, Xilin Jiang, Cong Han, and Nima Mesgarani. Styletts-zs: Efficient high-quality zero-shot text-to-speech synthesis with distilled time-varying style diffusion. *arXiv preprint arXiv:2409.10058*, 2024.

[24] Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. *arXiv preprint arXiv:2406.18009*, 2024.

[25] Dongchao Yang, Rongjie Huang, Yuanyuan Wang, Haohan Guo, Dading Chong, Songxiang Liu, Xixin Wu, and Helen Meng. Simplespeech 2: Towards simple and efficient text-to-speech with flow-based scalar latent transformer diffusion models. *arXiv preprint arXiv:2408.13893*, 2024.

[26] Keon Lee, Dong Won Kim, Jaehyeon Kim, and Jaewoong Cho. Ditto-tts: Efficient and scalable zero-shot text-to-speech with diffusion transformer. *arXiv preprint arXiv:2406.11427*, 2024.

[27] Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. *arXiv preprint arXiv:2410.06885*, 2024.

[28] Yingahao Aaron Li, Rithesh Kumar, and Zeyu Jin. Dmospeech: Direct metric optimization via distilled diffusion model in zero-shot speech synthesis. *arXiv preprint arXiv:2410.11097*, 2024.

[29] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024.

[30] Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*, pages 2709–2720. PMLR, 2022.

[31] Edresson Casanova, Christopher Shulby, Eren Gölge, Nicolas Michael Müller, Frederico Santos De Oliveira, Arnaldo Candido Junior, Anderson da Silva Soares, Sandra Maria Aluisio, and Moacir Antonelli Ponti. Sc-glowtts: An efficient zero-shot multi-speaker text-to-speech model. *arXiv preprint arXiv:2104.05557*, 2021.

[32] Yihan Wu, Xu Tan, Bohan Li, Lei He, Sheng Zhao, Ruihua Song, Tao Qin, and Tie-Yan Liu. Adaspeech 4: Adaptive text to speech in zero-shot scenarios. *arXiv preprint arXiv:2204.00436*, 2022.

[33] Sang-Hoon Lee, Seung-Bin Kim, Ji-Hyun Lee, Eunwoo Song, Min-Jae Hwang, and Seong-Whan Lee. Hierspeech: Bridging the gap between text and speech by hierarchical variational inference using self-supervised representations for speech synthesis. *Advances in Neural Information Processing Systems*, 35:16624–16636, 2022.

[34] Dongchan Min, Dong Bok Lee, Eunho Yang, and Sung Ju Hwang. Meta-stylespeech: Multi-speaker adaptive text-to-speech generation. In *International Conference on Machine Learning*, pages 7748–7759. PMLR, 2021.

[35] Yinghao Aaron Li, Cong Han, and Nima Mesgarani. Styletts: A style-based generative model for natural and diverse text-to-speech synthesis. *arXiv preprint arXiv:2205.15439*, 2022.

[36] Hyeong-Seok Choi, Jinhyeok Yang, Juheon Lee, and Hyeongju Kim. Nansy++: Unified voice synthesis with neural analysis and synthesis. *arXiv preprint arXiv:2211.09407*, 2022.

[37] Jingyi Chen, Ju-Seung Byun, Micha Elsner, and Andrew Perrault. Reinforcement learning for fine-tuning text-to-speech diffusion models. *arXiv preprint arXiv:2405.14632*, 2024.

[38] Xiaohui Sun, Ruitong Xiao, Jianye Mo, Bowen Wu, Qun Yu, and Baoxun Wang. F5r-tts: Improving flow matching based text-to-speech with group relative policy optimization. *arXiv preprint arXiv:2504.02407*, 2025.

[39] Hubert Siuzdak. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. *arXiv preprint arXiv:2306.00814*, 2023.

[40] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024.

[41] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

[43] Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 885–890, 2024.

[44] Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, Mingqing Gong, Peisong Huang, Qingqing Huang, Zhiying Huang, Yuanyuan Huo, Dongya Jia, Chumin Li, Feiya Li, Hui Li, Jiaxin Li, Xiaoyang Li, Xingxing Li, Lin Liu, Shouda Liu, Sichao Liu, Xudong Liu, Yuchen Liu, Zhengxi Liu, Lu Lu, Junjie Pan, Xin Wang, Yuping Wang, Yuxuan Wang, Zhengnan Wei, Jian Wu, Chao Yao, Yifeng Yang, Yuan-Qiu-Qiang Yi, Junteng Zhang, Qidi Zhang, Shuo Zhang, Wenjie Zhang, Yang Zhang, Zilin Zhao, Dejian Zhong, and Xiaobin Zhuang. Seed-tts: A family of high-quality versatile speech generation models. *ArXiv*, abs/2406.02430, 2024.

[45] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

[46] Tingwei Guo, Cheng Wen, Dongwei Jiang, Ne Luo, Ruixiong Zhang, Shuaijiang Zhao, Wubo Li, Cheng Gong, Wei Zou, Kun Han, et al. Didispeech: A large scale mandarin speech corpus. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6968–6972. IEEE, 2021.

[47] Ilya Loshchilov and Frank Hutter. Fixing Weight Decay Regularization in Adam, 2018.

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

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

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

[51] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[52] Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin P. Murphy, and Tim Salimans. Diffusion meets flow matching: Two sides of the same coin. 2024.

[53] Zhijun Liu, Shuai Wang, Sho Inoue, Qibing Bai, and Haizhou Li. Autoregressive diffusion transformer for text-to-speech synthesis. *arXiv preprint arXiv:2406.05551*, 2024.

[54] Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian. Wespeaker: A research and production oriented speaker embedding learning toolkit. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

A Additional Analyses

A.1 Impact of Duration Prediction on Speech Quality

Table 3: Impact of different duration prediction approaches on speech quality metrics. All evaluations are conducted on *Seed-TTS-en* dataset using the same speech generation model.

Duration Source	SIM↑	WER↓
Ground Truth Audio	0.734	2.143
Ground Truth Duration	0.697	1.821
Speaking Rate Based	0.682	2.028
Duration Predictor	0.672	3.750
Best-of-8 Sampling	0.724	1.723
DMOSpeech 2 (Ours)	0.698	1.752

Duration prediction plays a crucial role in non-autoregressive TTS systems, directly affecting both intelligibility and speaker similarity. To illustrate this impact, we conducted experiments comparing different duration determination methods on the *Seed-TTS-en* evaluation set. Table 3 presents the results.

We evaluated several approaches to determine speech duration: Ground Truth Audio refers to the original recordings; Ground Truth Duration uses reference durations from the dataset; Speaking Rate Based implements the F5-TTS approach of interpolating duration based on speaking rate; Duration Predictor shows results without RL optimization; Best-of-8 Sampling selects the best result from 8 different duration samples based on quality metrics; and DMOSpeech 2 features our proposed RL-optimized duration predictor.

The results demonstrate several important findings. First, the unoptimized duration predictor performs notably worse than other approaches, particularly in terms of intelligibility (WER of 3.750). This confirms our hypothesis that duration prediction is a critical bottleneck in TTS quality, which has also been shown in previous studies [24].

Second, the Best-of-8 sampling approach achieves the best results with a WER of 1.723 and SIM of 0.724. This represents an "oracle" upper bound on what could be achieved through effective duration prediction, as it leverages privileged information about outcome quality that would not be available during standard inference. This ceiling indicates the theoretical limit of what our RL approach could achieve with perfect optimization.

Notably, our proposed RL-optimized duration predictor (DMOSpeech 2) achieves a WER of 1.752, which is better than using ground truth durations (WER of 1.821) while maintaining competitive similarity (0.698). This demonstrates that our RL-based optimization successfully learns to predict durations that enhance speech intelligibility without requiring ground truth information. While not quite reaching the ceiling established by Best-of-8 sampling, our approach comes remarkably close while being significantly more efficient, requiring only a single forward pass during inference.

Interestingly, while using ground truth durations provides good intelligibility, it does not maximize speaker similarity (SIM of 0.697). This suggests that optimal durations for speaker similarity might differ slightly from those for intelligibility, highlighting the benefit of our joint optimization approach through reinforcement learning, which can balance these competing objectives.

A.2 Hyperparameter Selection for Duration Predictor RL

A.2.1 Group Size and Training Steps

To determine the optimal hyperparameters for our GRPO-based duration predictor training, we conducted extensive validation experiments using a small subset of the *Seed-TTS-en* evaluation set. Figure 4 illustrates the dynamics of model performance across different training steps and group sizes.

Our experiments revealed a critical training steps threshold around 1.5K steps, beyond which performance deteriorated significantly. With a group size of 8, both the speaker similarity and word error

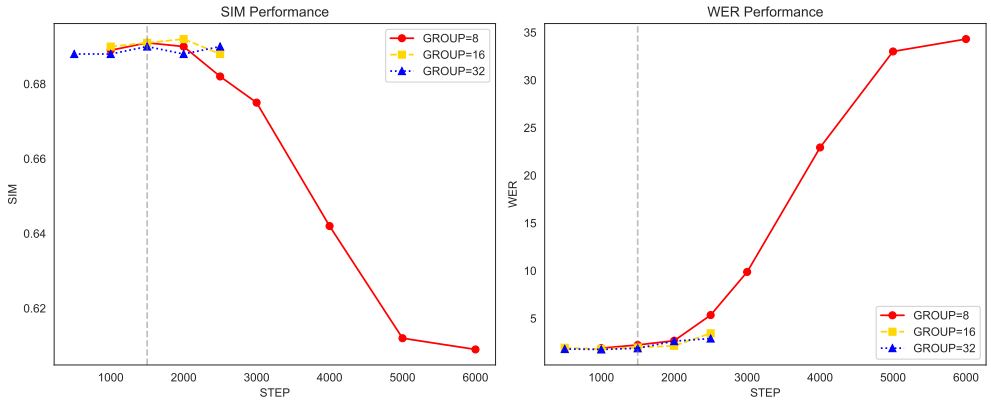

Figure 4: Performance dynamics during RL training of the duration predictor with various group sizes. The left plot shows speaker similarity (SIM) while the right plot shows word error rate (WER). The vertical dashed line indicates the 1.5K steps we selected for our final model.

rate metrics showed dramatic degradation after approximately 2K steps. This pattern suggests that extended training with reinforcement learning leads to overfitting to the reward signal, causing the policy to deviate excessively from the reference model.

Interestingly, while larger group sizes (16 and 32) demonstrated greater stability in performance over extended training, group size 16 emerged as the optimal configuration. We hypothesize that this superiority stems from group size 16 achieving an ideal balance between exploration and exploitation. With 16 samples per training instance, the model receives sufficient diversity in speech realizations to explore the duration space effectively, while maintaining enough focus on high-reward regions to exploit promising speech characteristics.

Additionally, group size 16 provides adequate statistical stability for reliable advantage estimation without introducing excessive computational overhead. Smaller groups (8) appear to suffer from high variance in advantage estimation, leading to unstable training, while larger groups (32) offer diminishing returns in performance improvement relative to their increased computational cost.

Based on these findings, we selected 1.5K training steps with a group size of 16 for our final model, which strikes an optimal balance between performance improvement and training efficiency. This configuration effectively improves the duration predictor's accuracy without deviating too far from the original supervised model, thereby avoiding the pitfalls of reward over-optimization.

A.2.2 Balancing Speaker Verification and Speech Recognition Rewards

A critical aspect of our reinforcement learning approach is properly balancing the contributions of speaker similarity and speech intelligibility in the reward function. Our reward formulation combines a speaker verification (SV) similarity term and a connectionist temporal classification (CTC) likelihood term:

$$r_k = \log p(\mathbf{x}|C(\mathbf{y}_k)) + \lambda_{\text{SIM}} \cdot \frac{\mathbf{e}_{\text{p}} \cdot \mathbf{e}_{y_k}}{\|\mathbf{e}_{\text{p}}\| \, \|\mathbf{e}_{y_k}\|}, \tag{10}$$

The selection of an appropriate λ_{SIM} value is crucial for ensuring that neither component dominates the optimization process. During our preliminary analysis, we observed that the CTC term ($\log p(\mathbf{x}|C(\mathbf{y}_k))$) typically produces values approximately three times larger in magnitude than the cosine similarity term with our CTC and SV models. This imbalance would naturally lead the duration predictor to prioritize intelligibility over speaker mimicry if left unaddressed.

To achieve a balanced optimization objective where both metrics contribute equally to model training, we conducted a series of calibration experiments. By analyzing the statistical distribution of both reward components across our validation set, we determined that setting $\lambda_{\text{SIM}} = 3$ effectively equalizes their contributions. This calibration ensures that improvements in speaker similarity receive comparable reinforcement to improvements in speech intelligibility.

Our experimental results confirm the effectiveness of this balanced approach. When using significantly lower values for λ_{SIM}, we observed that the model would converge to durations that produced more intelligible speech but with diminished speaker similarity. Conversely, with substantially higher values, the model prioritized speaker characteristics at the expense of comprehensibility. The selected value of $\lambda_{\text{SIM}} = 3$ achieves the optimal trade-off between these competing objectives, resulting in speech that maintains both high intelligibility and strong speaker similarity.

B DMOSpeech Technical Details

This section provides a comprehensive overview of the DMOSpeech framework [28] as adapted for flow matching models in DMOSpeech 2.

B.1 Flow Matching for Speech Synthesis

Our teacher model, F5-TTS [27], is based on the conditional flow matching (CFM) framework rather than the velocity prediction diffusion used in the original DMOSpeech. The flow matching objective is to match a probability path p_t from a simple distribution p_0 (standard normal) to a target distribution p_1 that approximates the data distribution q.

In the CFM framework, the model learns a vector field v_t that guides the transformation of samples from noise to data. The loss function is:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)} \|v_t((1-t)x_0 + tx_1) - (x_1 - x_0)\|^2 \tag{11}$$

where $t \sim \mathcal{U}[0,1]$ is the flow step, $x_0 \sim p(x_0)$ is sampled from the noise distribution, $x_1 \sim q(x_1)$ is sampled from the data distribution, and $(1-t)x_0 + tx_1$ represents the noisy sample at time t.

For speech synthesis, the input consists of a mel spectrogram $x_1 \in \mathbb{R}^{F \times N}$ where F is the mel dimension and N is the sequence length; a text embedding $\mathbf{c}$ derived from the input text; and a binary mask $\mathbf{m} \in \{0,1\}^{F \times N}$ that indicates which portions are prompt (to be preserved) and which are to be generated

The model v_t is trained to predict the flow vector field conditioned on these inputs. During training, we introduce a noisy sample $(1-t)x_0 + tx_1$ and the masked speech $(1-\mathbf{m}) \odot x_1$, where x_0 is Gaussian noise.

During inference, we use an ordinary differential equation (ODE) solver to transform noise x_0 into a mel spectrogram x_1 by integrating along the vector field:

$$\frac{d\psi_t(x_0)}{dt} = v_t(\psi_t(x_0)|\mathbf{c}, \mathbf{m}) \tag{12}$$

where $\psi_0(x_0) = x_0$ and we aim to compute $\psi_1(x_0) = x_1$.

B.2 Sway Sampling for Improved Inference

F5-TTS [27] introduced sway sampling to improve the efficiency and quality of speech generation. The sway sampling function is defined as:

$$f_{\text{sway}}(u; s) = u + s \cdot (\cos(\frac{\pi}{2}u) - 1 + u) \tag{13}$$

where $u \sim \mathcal{U}[0,1]$ and s is a coefficient controlling the sampling bias. This function transforms uniform samples to focus more on certain flow regions.

In DMOSpeech 2, we use a specific sway sampling schedule with the coefficient $s = -1$ that transforms our standard 4-step schedule $\{0.0, 0.25, 0.5, 0.75\}$ to $\{0.0000, 0.0761, 0.2929, 0.6173\}$ following [27]. This places more emphasis on the early steps of the generation process, allowing the model to establish better content and speaker foundations before refining details.

B.3 Distribution Matching Distillation

DMOSpeech 2 adapts the Improved Distribution Matching Distillation (DMD 2) framework [40] for flow matching models. The objective is to train a student generator G_θ to produce samples whose distribution matches the data distribution after applying the forward flow process.

We minimize the Kullback-Liebler (KL) divergence between the distributions of the sampled real data $p_{\text{data},t}$ and the sampled student generator outputs $p_{\theta,t}$ across all time $t \in [0,1]$:

$$\begin{aligned} D_{KL}(p_{\theta,t}||p_{\text{data},t}) &= \mathbb{E}_{\mathbf{x}\sim p_{\theta,t}}\left[\log\left(\frac{p_{\theta,t}(\mathbf{x})}{p_{\text{data},t}(\mathbf{x})}\right)\right] \\ &= -\mathbb{E}_{\mathbf{x}\sim p_{\theta,t}}\left[\log\left(p_{\text{data},t}(\mathbf{x})\right) - \log\left(p_{\theta,t}(\mathbf{x})\right)\right] \end{aligned} \tag{14}$$

The DMD loss is defined as:

$$\mathcal{L}_{\text{DMD}} = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\left[D_{KL}(p_{\theta,t}||p_{\text{data},t})\right] \tag{15}$$

For flow matching models, we adapt the gradient formulation [53]:

$$\nabla_\theta \mathcal{L}_{\text{DMD}} = -\mathop{\mathbb{E}}_{t,\mathbf{x}_t,\mathbf{z}}\left[\omega_t\left(v_{\text{real}}(\mathbf{x}_t,t) - v_\theta(\mathbf{x}_t,t)\right)\frac{dG}{d\theta}\right] \tag{16}$$

where $\mathbf{x}_t = (1-t)G_\theta(\boldsymbol{z}) + t\mathbf{x}_1$ for $\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, and v_{real} and v_θ are the vector fields from the teacher and student models, respectively. The weighting factor ω_t is defined as:

$$\omega_t = (1-t) \tag{17}$$

which gives more weight to earlier flow steps, aligning with the sway sampling philosophy.

B.4 Multi-step Sampling for Student Models

To address artifacts resulting from the one-step student model, we adapt the multi-step sampling approach from DMD 2 to the flow-matching model. The student generator G_θ is conditioned on the flow step t to estimate the mel spectrogram from a noisy counterpart at predefined time steps $t \in \{t_1,\ldots,t_N\}$.

The multi-step sampling algorithm follows:

Algorithm 3 DMD Multi-Step Sampling with Flow Matching

Require: Generator G_θ, flow steps $\{t_1,\ldots,t_N\}$, text embedding $\mathbf{c}$, prompt mask $\mathbf{m}$
1: Sample $\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})$
2: $\mathbf{x}_{t_1} \leftarrow \mathbf{z}$
3: **for** $n = 1$ to $N-1$ **do**
4: $\quad\hat{\mathbf{x}}_1^n \leftarrow G_\theta(\mathbf{x}_{t_n};\mathbf{c},\mathbf{m},t_n)$
5: $\quad$Sample $\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})$
6: $\quad\mathbf{x}_{t_{n+1}} \leftarrow (1-t_{n+1})\mathbf{z} + t_{n+1}\hat{\mathbf{x}}_1^n$
7: **end for**
8: $\hat{\mathbf{x}}_1^N \leftarrow G_\theta(\mathbf{x}_{t_N};\mathbf{c},\mathbf{m},t_N)$
9: **return** $\hat{\mathbf{x}}_1^N$

This creates a progressive refinement process, where earlier steps establish the content and speaker characteristics while later steps add details.

B.5 Multimodal Adversarial Training

To further improve the student model's performance, we incorporate adversarial training following **(author?)** [40]. Our discriminator D is a conformer that takes as input the stacked features from all transformer layers of the student network with noisy input, along with the text embeddings $\mathbf{c}$, prompt mask $\mathbf{m}$, and flow step t (denoted collectively as $\mathcal{C}$), adapted from [23].

The adversarial loss functions are:

$$\mathcal{L}_{\text{adv}}(G_\theta;D) = \mathbb{E}_{t,\hat{\mathbf{x}}_t\sim p_{\theta,t},\mathbf{m}}\left[\left(D\left(\hat{\mathbf{x}}_t;\mathcal{C}\right)-1\right)^2\right] \tag{18}$$

$$\begin{aligned} \mathcal{L}_{\text{adv}}(D;G_\theta) = \mathbb{E}_t\left[\mathbb{E}_{\hat{\mathbf{x}}_t\sim p_{\theta,t},\mathbf{m}}\left[\left(D\left(\hat{\mathbf{x}}_t;\mathcal{C}\right)\right)^2\right]\right] + \\ \mathbb{E}_t\left[\mathbb{E}_{\mathbf{x}_t\sim p_{\text{data},t},\mathbf{m}}\left[\left(D\left(\mathbf{x}_t;\mathcal{C}\right)-1\right)^2\right]\right] \end{aligned} \tag{19}$$

where $\mathcal{C} = \{\mathbf{c},\mathbf{m},t\}$ and $\hat{\mathbf{x}}_t = (1-t)\mathbf{z} + tG_\theta(\mathbf{z};\mathcal{C})$ for $\mathbf{z}\sim\mathcal{N}(0,I)$.

B.6 Direct Metric Optimization

DMOSpeech 2 retains the direct metric optimization approach from the original DMOSpeech, allowing end-to-end optimization of perceptual metrics. We directly optimize both speaker embedding cosine similarity (SIM) and word error rate (WER).

For WER improvement, we incorporate a connectionist temporal classification (CTC) loss:

$$\mathcal{L}_{\text{CTC}} = \mathbb{E}_{\mathbf{x}_{\text{fake}} \sim p_\theta, \mathbf{c}} \left[-\log p(\mathbf{c} | C(\mathbf{x}_{\text{fake}})) \right] \tag{20}$$

where $\mathbf{x}_{\text{fake}}$ is the student-generated mel spectrogram, $\mathbf{c}$ is the text transcript, and $C(\cdot)$ is a pre-trained CTC-based ASR model operating on mel-spectrograms.

For speaker similarity, we use a speaker verification (SV) loss:

$$\mathcal{L}_{\text{SV}} = \mathbb{E}_{\substack{\mathbf{x}_{\text{real}} \sim p_{\text{data}}, \\ \mathbf{x}_{\text{fake}} \sim p_\theta, \mathbf{m}}} \left[1 - \frac{\mathbf{e}_{\text{real}} \cdot \mathbf{e}_{\text{fake}}}{\|\mathbf{e}_{\text{real}}\| \, \|\mathbf{e}_{\text{fake}}\|} \right] \tag{21}$$

where $\mathbf{e}_{\text{real}} = S(\mathbf{x}_{\text{real}})$ and $\mathbf{e}_{\text{fake}} = S(\mathbf{x}_{\text{fake}})$ are the speaker embeddings of the prompt and student-generated speech, obtained from a pre-trained speaker verification model S.

B.7 Training Objectives and Stability

The overall training objective for G_θ combines DMD, adversarial, SV, and CTC losses:

$$\min_\theta \; \mathcal{L}_{\text{DMD}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}(G_\theta; D) + \lambda_{\text{SV}} \mathcal{L}_{\text{SV}} + \lambda_{\text{CTC}} \mathcal{L}_{\text{CTC}} \tag{22}$$

The training objectives for the student vector field model $g_{\boldsymbol{\psi}}$ and discriminator D are:

$$\min_{\boldsymbol{\psi}} \; \mathcal{L}_{\text{CFM}} \left(g_{\boldsymbol{\psi}}; p_\theta \right) \tag{23}$$

$$\min_D \; \mathcal{L}_{\text{adv}} \left(D; G_\theta \right) \tag{24}$$

We employ an alternating training strategy where G_θ, $g_{\boldsymbol{\psi}}$, and D are updated at different rates to maintain stability. For every update of G_θ, five updates of $g_{\boldsymbol{\psi}}$ are performed to ensure the vector field model can adapt quickly to changes in the generator distribution. The discriminator D and generator G_θ are updated at the same rate.

For training stability, following [28], the weights are set as follows: $\lambda_{\text{adv}} = 10^{-3}$ to balance the gradient norms, $\lambda_{\text{CTC}} = 0$ for the first 5,000 iterations, then $\lambda_{\text{CTC}} = 1$, and $\lambda_{\text{SV}} = 0$ for the first 10,000 iterations, then $\lambda_{\text{SV}} = 1$. This phased approach allows the generator to first learn basic speech generation before focusing on specific quality metrics.

B.8 Vocoder

Same as F5-TTS [27], DMOSpeech 2 uses the Vocos neural vocoder [39] to convert mel-spectrograms to waveforms. Vocos is a GAN-based vocoder that offers high-quality synthesis with efficient inference. The vocoder is pre-trained on a diverse dataset of speech recordings and is used as-is without fine-tuning during DMOSpeech 2 training and inference.

B.9 Automatic Speech Recognition (ASR) Model

The ASR model used for the CTC loss is a 6-layer transformer encoder trained directly on mel-spectrograms. The model is trained on Emilia using the CTC loss to align the speech with the text transcriptions for both Chinese and English.

B.10 Speaker Verification (SV) Model

The speaker verification model is a 6-layer transformer encoder with an additional projection layer that produces fixed-dimensional speaker embeddings. The model is distilled from the WeSpeaker [54] SimAMResNet34 model on the Emilia dataset following [28].

C Subjective Evaluation

In addition to the absolute rating evaluation described previously, we conducted comparative mean opinion score (CMOS) tests to directly assess the relative performance of our proposed models against baseline systems. As shown in Figure 5, the evaluation interface presents participants with three audio samples: a reference recording (top) and two synthesized speech samples (bottom) labeled as "Audio 1" and "Audio 2."

For each comparison, participants were instructed to:

1. Listen to all three audio samples
2. Select which of the two synthesized samples sounds more natural (left question)
3. Select which of the two synthesized samples sounds more similar to the reference voice (right question)

The DMOSpeech 2 model (with 4 sampling steps) served as the anchor system for all comparisons, with participants unaware of which sample corresponded to which system. The dropdown selection options were coded as follows: a rating of 0 indicates no preference, positive values indicate a preference for Audio 2, and negative values indicate a preference for Audio 1. This design allows for direct assessment of relative differences between systems without requiring absolute judgments on a fixed scale.

We collected responses from a total of 320 English and 320 Chinese samples. To ensure data quality, we employed validation checks similar to those in our previous evaluation, including mismatched speaker checks and identical sample pairs. Participants failing these validation tests were excluded from the final analysis. Statistical significance was determined using a paired t-test.

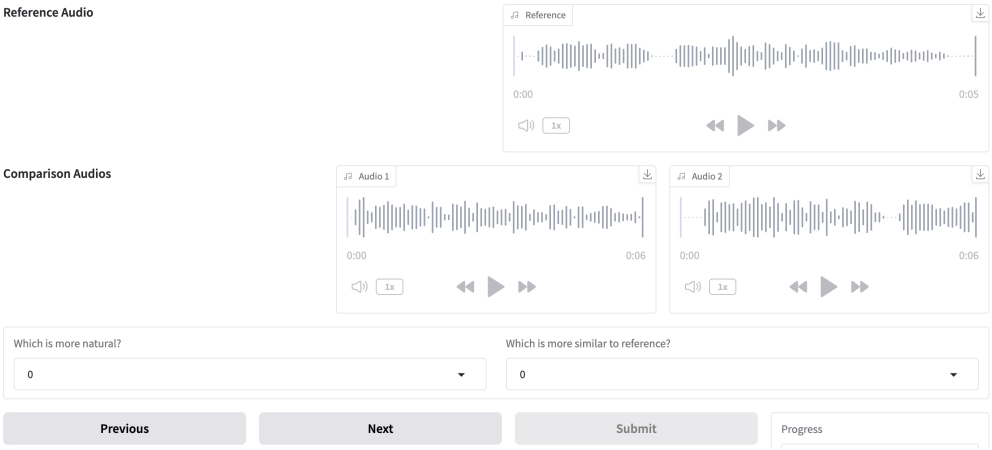

Figure 5: Screenshot of the comparative subjective evaluation interface. The interface presents three audio samples: a reference recording at the top, and two synthesized speech samples for comparison below. Participants are asked to make direct comparisons between the two synthesized samples by selecting which one sounds more natural and which one is more similar to the reference voice.