Advancing Zero-shot Text-to-Speech Intelligibility across Diverse Domains via Preference Alignment

Xueyao Zhang[*,1], **Yuancheng Wang**[*,1], **Chaoren Wang**[1],
Ziniu Li[1], **Zhuo Chen**[2], **Zhizheng Wu**[1]
[1]The Chinese University of Hong Kong, Shenzhen
[2]ByteDance Seed

Abstract

Modern zero-shot text-to-speech (TTS) systems, despite using extensive pre-training, often struggle in challenging scenarios such as tongue twisters, repeated words, code-switching, and cross-lingual synthesis, leading to intelligibility issues. To address these limitations, this paper leverages preference alignment techniques, which enable targeted construction of out-of-pretraining-distribution data to enhance performance. We introduce a new dataset, named the Intelligibility Preference Speech Dataset (INTP), and extend the Direct Preference Optimization (DPO) framework to accommodate diverse TTS architectures. After INTP alignment, in addition to intelligibility, we observe overall improvements including naturalness, similarity, and audio quality for multiple TTS models across diverse domains. Based on that, we also verify the weak-to-strong generalization ability of INTP for more intelligible models such as CosyVoice 2 and Ints. Moreover, we showcase the potential for further improvements through iterative alignment based on Ints. Audio samples are available at https://intalign.github.io/.

1 Introduction

Despite leveraging large-scale pre-training (Anastassiou et al., 2024; Wang et al., 2025a; Du et al., 2024b), modern zero-shot TTS systems still lack robustness during real-world applications (Sahoo et al., 2024; Neekhara et al., 2024). These systems struggle to meet even the most fundamental requirement of speech synthesis – *intelligibility* (Tan, 2023) in several scenarios, including: (1) the target text is hard to pronounce, such as tongue twisters or continuously repeated words (Neekhara et al., 2024; Anastassiou et al., 2024), which is referred to as *articulatory* cases in this paper, (2) *code-switching* cases, where the target text contains a mixture of multiple languages, and (3) *cross-lingual* cases, where the languages of the target text and the reference speech differ. In these domains, existing zero-shot TTS models frequently exhibit "hallucination" issues, such as content insertion, omission, and mispronunciation (Neekhara et al., 2024; Wang et al., 2023).

We attribute these intelligibility challenges primarily to the problem of *out-of-distribution* (OOD). For example, in cross-lingual cases, there exists a huge mismatch between monolingual pre-training and cross-lingual inference. While including such scenarios in pre-training data would be a natural solution, collecting high-quality data for challenging cases like cross-lingual synthesis remains difficult.

Motivated by the above, we propose to use *preference alignment* (PA) (Ouyang et al., 2022; Bai et al., 2022) to mitigate the OOD issues, and thus enhance zero-shot TTS intelligibility. The potential of this approach lies in two aspects. First, PA's *customized* post-training on human expected distribution can effectively mitigate the OOD issue (Zhang et al., 2024b; Li et al., 2024a; Xiong et al., 2024). Second, unlike TTS pre-training that requires high-quality supervised data, PA needs only paired samples with relative preferences – notably, even synthetic data can lead to large improvements (Dubey et al., 2024; Yang et al., 2024b), thus significantly simplifying data collection for challenging scenarios like cross-lingual cases. Centered on this direction, we investigate three research problems:

- **P1**: Dataset quality is crucial for model performance. To construct a high-quality intelligibility preference dataset, what prompts and base models should be selected, and how can we establish human-aligned preference pairs?
- **P2**: Unlike textual LLMs with predominantly autoregressive (AR) design, zero-shot TTS models employ diverse architectures, including AR-based (Borsos et al., 2023a; Anastassiou et al.,

[*]Equal Contribution.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12251–12270
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

2024; Du et al., 2024b), Flow-Matching (FM) based (Le et al., 2023; Eskimez et al., 2024; Chen et al., 2024c), and Masked Generative Model (MGM) based (Ju et al., 2024; Wang et al., 2025a). How can we design alignment algorithms for various architectures?

- **P3**: Can our preference dataset demonstrate weak-to-strong generalization (Burns et al., 2024)? In other words, can datasets created using less capable models effectively train more powerful models? This question is central to understanding the scalability and transferability of our dataset design.

In this paper, we address the aforementioned problems with the following key contributions:

→ **P1**: We establish a synthetic Intelligibility Preference Speech Dataset (INTP), comprising about 250K preference pairs (over 2K hours) of diverse domains. Specifically, INTP covers multiple scenarios, utilizing various TTS models for data creation. Besides, we employ several strategies to construct preference pairs, aiming to mitigate the risk of reward hacking for simple patterns (Skalse et al., 2022; Weng, 2024). Particularly, we leverage human knowledge and DeepSeek-V3 (DeepSeek-AI et al., 2024) to introduce perturbations into TTS systems, creating human-guided negative samples. In addition, when using Word Error Rate (WER) to determine intelligibility preferences, we not only consider self-comparison within a single model as in previous studies (Tian et al., 2024; Yao et al., 2025; Hussain et al., 2025), but also introduce comparisons across different models to leverage their complementary capabilities.

→ **P2**: We adopt the idea of Direct Preference Optimization (DPO) (Rafailov et al., 2023) to enhance various zero-shot TTS architectures. We employ the vanilla DPO algorithm for AR-based TTS models, while proposing extended versions of it for FM-based and MGM-based models. Our experiments on INTP shows that these algorithms effectively improve the intelligibility, naturalness, and overall quality of multiple state-of-the-art TTS systems, including ARS (AR-based) (Wang et al., 2025a), F5-TTS (FM-based) (Chen et al., 2024c), and MaskGCT (MGM-based) (Wang et al., 2025a).

→ **P3**: To investigate INTP's weak-to-strong generalization capability (Burns et al., 2024) on more powerful base models, we research its alignment effects on CosyVoice 2 (Du et al., 2024b) and Ints (Appendix C). Both models are initialized from textual LLMs (CosyVoice 2: from Qwen2.5, 0.5B (Yang et al., 2024a). Ints: from Phi-3.5-mini-instruct, 3.8B (Abdin et al., 2024)) and achieve superior intelligibility performance (Table 4). Our experimental results verify that INTP, though constructed from weaker models, remains effective for these two strong models. Additionally, we showcase how to establish an *iterative* preference alignment "flywheel" of data and model improvements (Bai et al., 2022; Dubey et al., 2024; Xiong et al., 2024) based on Ints.

We will open-source all resources used in this study at Amphion[1] (Zhang et al., 2024c), including: (1) the proposed INTP and DPO-based alignment codebase for various TTS models, (2) all the INTP-enhanced models based on Ints, CosyVoice 2, ARS, F5-TTS, and MaskGCT, and (3) our newly constructed zero-shot TTS evaluation sets across diverse domains.

2 Related Work

Zero-Shot Text to Speech Given a target text and a reference speech as input, zero-shot TTS systems aim to synthesize the target text while mimicking the reference style. Modern zero-shot TTS systems include AR approaches (Wang et al., 2023; Peng et al., 2024; Anastassiou et al., 2024; Guo et al., 2024; Du et al., 2024a,b; Zhang et al., 2025) that model discrete speech tokens (Zeghidour et al., 2021; Défossez et al., 2023), and Non-AR approaches that either model continuous representations using diffusion (Shen et al., 2024) or flow matching (Le et al., 2023; Eskimez et al., 2024; Chen et al., 2024c), or model discrete tokens using masked generative models (Borsos et al., 2023b; Ju et al., 2024; Wang et al., 2025a,b). While these systems, trained on large-scale datasets (He et al., 2024; Kahn et al., 2020; He et al., 2025), show excellent intelligibility in regular cases (Anastassiou et al., 2024; Panayotov et al., 2015; Du et al., 2024b), they still struggle with intelligibility in real-world scenarios.

Alignment for Speech Generation Alignment via post-training has demonstrated its effectiveness in the generation of text (Ouyang et al., 2022; Bai et al., 2022), vision (Xu et al., 2023; Fu et al., 2024), speech (Zhang et al., 2024a; Anastassiou et al., 2024; Du et al., 2024b), music (Cideron et al., 2024), and sound effects (Majumder et al., 2024; Liao et al., 2024). In speech generation, existing

[1] https://github.com/open-mmlab/Amphion

	Regular	Repeated	Code-Switching	Pronunciation-perturbed	Punctuation-perturbed	#Total
ARS (Wang et al., 2025a)	8,219	8,852	8,300	7,325	8,036	40,732
F5-TTS (Chen et al., 2024c)	8,425	8,555	7,976	7,909	6,667	39,532
MaskGCT (Wang et al., 2025a)	9,055	10,263	8,289	7,604	7,686	42,897
Intra Pairs	25,699	27,670	24,565	22,838	22,389	123,161
Inter Pairs	27,008	27,676	24,651	25,045	23,970	128,350
#Total	52,707	55,346	49,216	47,883	46,359	251,511

(a) Distribution of preference pairs, where pronunciation-perturbed and punctuation-perturbed texts are introduced to create the human-guided negative samples.

Text Type	Example
Regular	*A panda eats shoots and leaves.*
Repeated	*A panda panda eats shoots and leaves and leaves and leaves.*
Code-Switching	熊猫吃 *shoots* 和 *leaves*。
Pronunciation-perturbed	*A pan duh eights shots n leafs.*
Punctuation-perturbed	*A panda eats, shoots, and leaves.*

(b) Examples of different types for a text, "*A panda eats shoots and leaves*".

Table 1: Intelligibility Preference dataset (INTP). There are about 250K pairs (over 2K hours) in INTP, covering various texts and speechs, multiple models, and diverse preference pairs.

works have employed preference alignment to enhance multiple aspects of speech, including intelligibility (Anastassiou et al., 2024; Du et al., 2024b; Tian et al., 2024), speaker similarity (Anastassiou et al., 2024; Du et al., 2024b; Tian et al., 2024), emotion controllability (Anastassiou et al., 2024; Gao et al., 2024), and overall quality (Zhang et al., 2024a; Chen et al., 2024a; Hu et al., 2024; Chen et al., 2024b; Yao et al., 2025; Hussain et al., 2025). For intelligibility, previous studies choose WER as the optimization objective, either directly employing it as a reward model (Anastassiou et al., 2024; Du et al., 2024b) or centering around it to construct preference pairs (Tian et al., 2024; Yao et al., 2025; Hussain et al., 2025).

However, the existing research exhibits two main limitations. First, in constructing intelligibility preference dataset, current works rely solely on a single model to generate data (Tian et al., 2024; Yao et al., 2025; Hussain et al., 2025), neglecting comparisons across different models. Additionally, beyond the objective WER, the potential of leveraging human knowledge or feedback to construct preference pairs remains unexplored. Second, most existing work has focused primarily on optimizing AR-based (Zhang et al., 2024a; Anastassiou et al., 2024; Du et al., 2024b; Tian et al., 2024) or diffusion-based (Chen et al., 2024b) TTS models, leaving open the question of how to design effective alignment algorithms for other architectural paradigms, such as FM-based and MGM-based TTS models.

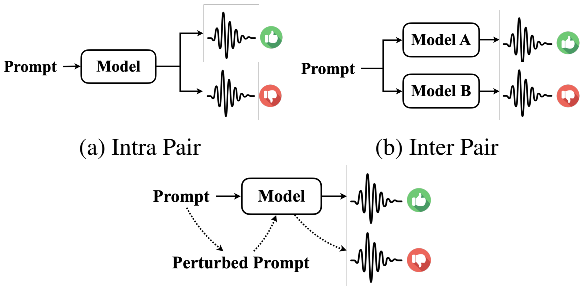
3 INTP: Intelligibility Preference Speech Dataset

To enhance the TTS intelligibility, this study opts for constructing a preference dataset to align (Tian et al., 2024; Yao et al., 2025; Hussain et al., 2025) rather than directly optimizing single metrics or rules such as WER (Anastassiou et al., 2024; Du et al., 2024b). This choice is motivated by two key considerations. First, through the construction of a preference dataset, we can inject human knowledge and feedback beyond WER, such as creating human-guided negative samples in the framework of preference alignment (Section 3.3). Second, in addition to the existing approach of constructing preference pairs from multiple samples of a single model (Tian et al., 2024; Yao et al., 2025; Hussain et al., 2025), we can leverage comparisons across different models to create preference pairs, thereby utilizing the complementary capabilities of various models (Figure 1b). These different strategies help increase diversity in the dataset, mitigating the risk of "reward hacking" that often results from the simple patterns inherent in single metrics or rules (Bai et al., 2022; Skalse et al., 2022; Weng, 2024).

Formally, we aim to construct an intelligibility preference dataset $\mathcal{D} = \{(x, y^w, y^l)\}$, where each triplet comprises a prompt x (consisting of target text x^{text} and reference speech x^{speech} for zero-shot TTS models), along with a pair of synthesized speech samples (y^w, y^l). Here, y^w and y^l represent the preferred (positive) and dispreferred (negative) outputs conditioned on x, respectively. Statistics of the proposed INTP are presented in Table 1.

3.1 Prompt Construction

To establish a high-quality preference dataset, we aim to make the distribution of prompt x cover a wide range of domains. For the target text x^{text}, from the linguistic perspective, we design three distinct categories: (1) **Regular text**, which represents the general cases for TTS systems, aimed at enhancing model intelligibility in common scenarios; (2) **Repeated text**, which contains repeated or redundant words and phrases, specifically designed to improve TTS performance in articulatory cases; and (3) **Code-switching text**, which incorporates a mixture of different languages, intended to en-



(a) Intra Pair (b) Inter Pair

(c) Perturbed Pair

Figure 1: Three kinds of preference pairs in INTP.

hance TTS capabilities in multilingual scenarios. From the semantic perspective, we collect text content across diverse topics and domains to enrich the distribution of x^{text}. For the reference speech x^{speech}, we aim to cover a wide range of speakers, speaking styles, and acoustic environments. Regarding the pairing of x^{text} and x^{speech}, we further consider their language alignment by constructing both **monolingual** and **cross-lingual** combinations (more statistics in Appendix A.1).

We construct these prompt data based on the Emilia-Large (He et al., 2024, 2025), which contains real-world speech data and textual transcriptions across diverse topics, scenarios, and speaker styles. We perform stratified sampling on Emilia-Large's speech and text data to obtain multilingual prompts. We employ DeepSeek-V3 (DeepSeek-AI et al., 2024) to preprocess the sampled text, including typo correction, and use it as regular text. Based on these regular texts, we further utilize DeepSeek-V3 to transform them into different text types (as shown in Table 1b). Construction details are provided in Appendix A.1.

3.2 Model Selection

We utilize multiple zero-shot TTS models with diverse architectures for data synthesis to enhance INTP's diversity and generalization. Specifically, we select the following three models: (1) **ARS** (AR-based): Introduced as an autoregressive baseline by Wang et al. (2025a). and referred to as "AR + SoundStorm" in the original paper (Wang et al., 2025a). It adopts a cascaded architecture, including the autoregressive *text-to-codec* and the non-autoregressive *codec-to-waveform* (Borsos et al., 2023b). (2) **F5-TTS** (FM-based): It follows E2 TTS (Eskimez et al., 2024) and uses a flow-matching transformer (Le et al., 2023; Lipman et al., 2023) to convert the text to acoustic features directly (Chen et al., 2024c). (3) **MaskGCT** (MGM-based): Similar to ARS, MaskGCT employs a two-stage architecture. The key distinction lies in its use of an MGM in the text-to-codec stage (Wang et al., 2025a).

All the three are pre-trained on Emilia (He et al., 2024) (about 100K hours of multilingual data) and represent state-of-the-art zero-shot TTS systems across different architectures. We utilize their officially released pre-trained models (see Appendix A.2 for details) to generate data for INTP.

3.3 Preference Pairs Construction

In constructing intelligibility preference pairs, we design three categories of pairs (Figure 1):

Intra Pair These pairs are generated through model self-comparison (Figure 1a), following an approach similar to previous studies (Tian et al., 2024; Yao et al., 2025; Hussain et al., 2025). For a given prompt x, we conduct multiple samplings using the same model. Subsequently, we calculate the WER for each generation and designate the samples with the lowest and highest WER as y^w and y^l, respectively. To enlarge the gap between y^w and y^l, we employ diverse sampling hyperparameters across multiple generations from the same model. Additionally, we use a specific WER threshold to filter out pairs with insufficient performance gaps (more details in Appendix A.3.1).

Inter Pair These pairs are constructed by comparing outputs across different models (Figure 1b). The efficacy of this approach lies in leveraging the complementary strengths of various models. For example, by comparing intra-pairs from different models for the same prompt, we can identify the "best of the best" samples, thereby enhancing the overall quality of positive samples in our dataset. Similar to intra pair, we also employ WER to identify intelligibility preferences for inter pairs (see Appendix A.3.2 for details).

Notably, the proposed inter-pair construction pipeline enables comparative evaluation of intelligibility performance across different models. Using this pipeline, we compared four state-of-the-art models in the field: ARS (Wang et al., 2025a), F5-TTS (Chen et al., 2024c), MaskGCT (Wang et al., 2025a), and CosyVoice 2 (Du et al., 2024b). We constructed 10K inter-pairs and analyzed the win rates of these models, as shown in Table 2. Interestingly, even ARS, the model with the lowest win rate, achieves a 4.1% success rate against the strongest model, CosyVoice 2. This finding validates our assumption regarding the complementary

	ARS	F5-TTS	MaskGCT	CosyVoice 2	Win Rate (↑)
ARS	/	6.7%	7.4%	4.1%	18.3%
F5-TTS	10.4%	/	8.8%	5.9%	25.1%
MaskGCT	10.4%	8.0%	/	5.9%	24.3%
CosyVoice 2	11.9%	10.2%	10.3%	/	32.3%

* The percentage in each cell represents the proportion of cases where the model on the horizontal axis outperforms the model on the vertical axis.
* The **Win Rate** is calculated as the sum of values from columns 2 through 5.

Table 2: TTS Intelligibility Arena: We employ the inter-pair construction from INTP to compare intelligibility among four state-of-the-art zero-shot TTS models.

	ARS	F5-TTS	MaskGCT	CosyVoice 2
Positive Samples	73.0%	88.1%	90.9%	100.0%
Negative Samples	45.7%	15.8%	47.1%	75.0%
All	59.7%	53.7%	64.3%	90.4%

Table 3: Human-annotated reading accuracy (↑) for four state-of-the-art zero-shot TTS models on regular texts. We use the intra-pair pipeline of INTP to generate the positive and negative samples.

capabilities among various models.

Perturbed Pair In addition to the aforementioned two types of pairs which are established based on WER, we leverage human knowledge and the intelligence of DeepSeek-V3 (DeepSeek-AI et al., 2024) to create human-guided negative samples, termed perturbed pairs (Figure 1c). The main idea involves deliberately perturbing the input prompt, thereby inducing the model to generate low-quality samples (Majumder et al., 2024; Fu et al., 2024).

Specifically, we design two types of perturbation for the target text in the prompt (as shown in Table 1b): (1) **Pronunciation perturbation**: we replace certain characters of the text with easily mispronounceable alternatives. For example, given the text "*A panda eats shoots and leaves*", we can create the perturbed text "*A pan duh eights shots n leafs*". (2) **Punctuation perturbation**: we modify the punctuation, such as commas, to alter pause patterns and prosody in the text. For example, by adding commas to the text "*A panda eats shoots and leaves*", we obtain "*A panda eats, shoots, and leaves*", where the words "*shoots*" and "*leaves*" transform from nouns in the original text to verbs, creating a significant semantic shift. The detailed process for constructing these perturbed texts is provided in Appendix A.3.3.

3.4 Human Perception Verification

After constructing INTP, we further conducted subjective evaluation to verify its alignment with human perception. For intelligibility alignment, we design a **reading accuracy** listening task (see Appendix F.3 for details): given a text and a speech, subjects perform binary classification to determine whether the speech accurately reads the text without any content insertion, omission, or mispronunciation. Using four state-of-the-art zero-shot TTS models, we generate 300 intra-pairs on INTP regular texts. The results in Table 3 demonstrate that INTP's preference identification for intra pairs aligns well with human judgments of intelligibility. Furthermore, comparing Tables 2 and 3 reveals that INTP's inter-pair comparisons of intelligibility across different models also effectively align with human values.

In addition to intelligibility, we also investigated how well INTP aligns with human preferences for *naturalness*, which is one of the most general-purpose metrics for TTS (Tan, 2023). The experimental results demonstrate that the naturalness gap between positive and negative samples of INTP is substantial and perceptible to human listeners. We discuss this finding in details in Appendix A.4.

4 Preference Alignment for Diverse Zero-Shot TTS models

In this section, we present methods for achieving preference alignment across a range of TTS models, including autoregressive based, flow-matching based, and masked generative model based architectures. Building on the framework of Direct Preference Optimization (DPO) (Rafailov et al., 2023), initially developed for AR-based models, we adapt and extend its principles to FM-based and MGM-based models. We note that DPO is computationally efficient in practice, and its iterative variant aligns seamlessly with the online reinforcement learning (RL) framework (Li et al., 2024b).

4.1 DPO for AR Models

The main idea of reinforcement learning (RL) for preference alignment is to introduce a reward model $r(x, y)$ to guide the model for improvement (see e.g., (Li et al., 2024b)). Here y represents the output (i.e., the generated speech in zero-shot TTS), and x means the input prompt (i.e., the reference speech and the target text in zero-shot TTS). A widely adopted reward model design is based on Bradley-Terry (BT) model, which defines the probability of preferred sample y^w over dispreferred sample y^l given x as $p_{\mathrm{BT}}(y^w \succ y^l \mid x) =$

$\sigma(r(x,y^w)-r(x,y^l))$. We can train the reward model $r_\phi(x,y)$ by minimizing the negative log-likelihood of observed comparisons from the preference dataset $\mathcal{D}$:

$$\mathcal{L}_{\text{R}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(r_\phi(x,y_w)-r_\phi(x,y_l)\right)\right]. \tag{1}$$

With the given reward model, the RL optimization objective is to guide the model to maximize the expected reward while minimizing the KL-divergence from a reference distribution:

$$\max_{p_\theta} \mathbb{E}_{x,y\sim p_\theta(y|x)}[r(x,y)] - \beta D_{\text{KL}}[p_\theta(y|x) \parallel p_{\text{ref}}(y|x)], \tag{2}$$

where the hyperparameter β controls the strength of the regularization. As highlighted in Rafailov et al. (2023), the optimization problem in Equation 2 admits a closed form solution. This implies a direct relationship between the reward function and the policy. Substituting the reward expression into Equation 1 leads the DPO loss:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{\mathcal{D}}\left[\log\sigma\left(\beta\left(\log\frac{p_\theta(y_w|x)}{p_{\text{ref}}(y_w|x)}-\log\frac{p_\theta(y_l|x)}{p_{\text{ref}}(y_l|x)}\right)\right)\right]. \tag{3}$$

DPO enables direct preference alignment for AR-based TTS models, eliminating the need for explicit reward modeling or RL optimization. In the following subsections, we will introduce its extensions for FM-based and MGM-based TTS models.

4.2 DPO for Flow-Matching Models

The vanilla DPO algorithm is tailored for AR models, while Wallace et al. (2024) extends it to diffusion models. In this subsection, we introduce the DPO algorithm for flow-matching models, specifically demonstrating its application to optimal transport flow-matching (OT-FM), a common approach in FM-based TTS models (Le et al., 2023; Eskimez et al., 2024; Chen et al., 2024c). Given the continuous representation y of a speech sample and its corresponding condition x, OT-FM constructs a linear interpolation path between Gaussian noise $y_0\sim\mathcal{N}(0,I)$ and the target data $y_1=y$. Specifically, the interpolation follows $y_t=(1-t)y_0+t\,y_1$, where $t\in[0,1]$, which naturally induces a velocity field $v_\theta(y_t,t,x)$ that captures the constant directional derivative $\frac{dy_t}{dt}=y_1-y_0$. OT-FM aims to learn the velocity field to match the true derivative. The corresponding loss function is defined as

$$\mathcal{L}_{\text{OT-FM}} = \mathbb{E}_{y_0,y_1,x,t}\| v_\theta(y_t,t,x)-(y_1-y_0)\|_2^2, \tag{4}$$

where t is the time step that is sampled from the uniform distribution $\mathcal{U}(0,1)$.

Inspired by Wallace et al. (2024), we rewrite the RL objective for flow-matching models. Let $p_\theta(y_1|y_t,t,x)$ denote our policy that predicts the target sample y_1 given the noised observation y_t at time t and condition x. We initialize from a reference flow-matching policy p_{ref}. The RL objective can be written as:

$$\begin{aligned}&\max_{p_\theta}\mathbb{E}_{y_1\sim p_\theta(y_1|x),t,x}[r(y_1,x)]\\&\quad-\beta\mathbb{D}_{\text{KL}}[p_\theta(y_1|y_t,t,x)\|p_{\text{ref}}(y_1|y_t,t,x)].\end{aligned} \tag{5}$$

Following a similar derivation process as in DPO (we provide more details in Appendix B.2), we can obtain the loss function for flow-matching DPO:

$$\begin{aligned}&\mathcal{L}_{\text{DPO-FM}} = -\mathbb{E}_{(y_1^w,y_1^l,x)\sim\mathcal{D},t}\\&\log\sigma\left(\beta\left(\log\frac{p_\theta(y_1^w|y_t^w,t,x)}{p_{\text{ref}}(y_1^w|y_t^w,t,x)}-\log\frac{p_\theta(y_1^l|y_t^l,t,x)}{p_{\text{ref}}(y_1^l|y_t^l,t,x)}\right)\right),\end{aligned} \tag{6}$$

where y_1^w and y_1^l represent the preferred and dispreferred samples from the preference dataset, respectively, while y_t^w and y_t^l are the interpolations at time t between y_1^w and y_1^l and the randomly sampled y_0^w and y_0^l. The loss can be transformed into the velocity space:

$$\begin{aligned}&\mathcal{L}_{\text{DPO-FM}} = -\mathbb{E}_{(y_1^w,y_1^l,x)\sim\mathcal{D},t}\log\sigma\Big(-\beta\\&\left(\|v_\theta(y_t^w,t,x)-(y_1^w-y_0^w)\|_2^2-\|v_{\text{ref}}(y_t^w,t,x)-(y_1^w-y_0^w)\|_2^2\right)\\&-\left(\left\|v_\theta(y_t^l,t,x)-(y_1^l-y_0^l)\right\|_2^2-\left\|v_{\text{ref}}(y_t^l,t,x)-(y_1^l-y_0^l)\right\|_2^2\right)\Big).\end{aligned} \tag{7}$$

This proposed algorithm can be applied to a wide range of FM-based and diffusion-based TTS models (Le et al., 2023; Eskimez et al., 2024; Shen et al., 2024). In this study, we use it to optimize F5-TTS (Chen et al., 2024c) as a representative.

4.3 DPO for Masked Generative Models

Masked generative model (MGM) is a type of Non-AR generative model, which is also widely adopted in speech generation, as seen in models such as NaturalSpeech 3 (Ju et al., 2024), and MaskGCT (Wang et al., 2025a). MGM aims to recover a discrete sequence $y=[z_1,z_2,\ldots,z_n]$ from its partially masked version $y_t=y\odot m_t$, where $m_t\in\{0,1\}^n$ is a binary mask sampled via a schedule $\gamma(t)\in(0,1]$. MGM is trained to predict masked tokens from unmasked tokens and condition x, modeled as $p_\theta(y_0\mid y_t,x)$, optimizing the sum of the marginal cross-entropy for each unmasked token:

$$\mathcal{L}_{\text{mask}} = -\mathbb{E}_{y,x,t,m_t}\sum_{i=1}^{n} m_{t,i}\cdot\log p_\theta(z_i\mid y_t,x). \tag{8}$$

Using a similar derivation as in Section 4.2, we extend DPO for MGM. Let $p_{\text{ref}}(y_0\mid y_t,x)$ represent

Model	Regular cases			Articulatory cases			Code-switching cases			Cross-lingual cases			Avg		
	WER	SIM	N-CMOS	WER	SIM	N-CMOS	WER	SIM	N-CMOS	WER	SIM	N-CMOS	WER	SIM	N-CMOS
ARS	3.96	0.717	-	20.03	0.693	-	54.15	0.693	-	19.76	0.630	-	24.47	0.683	-
w/ INTP	2.32	0.727	0.47 $_{\pm 0.22}$	12.83	0.713	0.64 $_{\pm 0.31}$	36.91	0.698	0.63 $_{\pm 0.34}$	9.57	0.632	0.82 $_{\pm 0.28}$	15.41	0.692	0.64 $_{\pm 0.12}$
F5-TTS	3.44	0.670	-	16.84	0.635	-	33.99	0.609	-	16.86	0.546	-	17.78	0.615	-
w/ INTP	2.38	0.652	0.38 $_{\pm 0.26}$	12.97	0.628	0.30 $_{\pm 0.23}$	15.98	0.576	0.67 $_{\pm 0.36}$	7.13	0.509	0.47 $_{\pm 0.30}$	9.62	0.591	0.44 $_{\pm 0.12}$
MaskGCT	2.34	0.738	-	12.43	0.714	-	29.06	0.696	-	12.34	0.629	-	14.04	0.694	-
w/ INTP	2.23	0.737	0.23 $_{\pm 0.20}$	9.13	0.722	0.57 $_{\pm 0.36}$	19.70	0.704	0.19 $_{\pm 0.16}$	7.87	0.633	0.29 $_{\pm 0.18}$	9.73	0.699	0.32 $_{\pm 0.15}$
CosyVoice 2	2.09	0.709	-	8.12	0.696	-	33.36	0.672	-	8.78	0.600	-	13.09	0.669	-
w/ INTP	1.65	0.709	0.24 $_{\pm 0.25}$	6.87	0.696	0.20 $_{\pm 0.16}$	28.31	0.671	0.63 $_{\pm 0.30}$	5.39	0.603	0.28 $_{\pm 0.31}$	10.56	0.670	0.33 $_{\pm 0.12}$
Ints	3.14	0.688	-	12.08	0.666	-	22.88	0.646	-	9.78	0.572	-	11.97	0.643	-
w/ INTP	2.36	0.686	0.20 $_{\pm 0.36}$	9.38	0.664	0.11 $_{\pm 0.22}$	13.80	0.642	0.20 $_{\pm 0.38}$	6.28	0.571	0.18 $_{\pm 0.23}$	7.96	0.641	0.17 $_{\pm 0.15}$

Table 4: Improvements of DPO with INTP for different models (**AR-based**: ARS (Wang et al., 2025a), CosyVoice 2 (Du et al., 2024a), and Ints (Appendix C). **FM-based**: F5-TTS (Chen et al., 2024c). **MGM-based**: MaskGCT (Wang et al., 2025a)) on diverse domains. ARS, F5-TTS, and MaskGCT participated in the INTP construction, while CosyVoice 2 and Ints did not.

the reference policy. The DPO loss for MGM is given by:

$$\mathcal{L}_{\text{DPO-MGM}} = -\mathbb{E}_{(y^w, y^l, x)\sim\mathcal{D}, t} \log\sigma\left(\beta\left(\log\frac{p_\theta(y_0^w|y_t^w, x)}{p_{\text{ref}}(y_0^w|y_t^w, x)} - \log\frac{p_\theta(y_0^l|y_t^l, x)}{p_{\text{ref}}(y_0^l|y_t^l, x)}\right)\right). \tag{9}$$

Here, y_t^w and y_t^l are masked versions of y_0^w and y_0^l. Note that $p_\theta(y_0|y_t, x)$ corresponds to the sum of the log-probabilities of the unmasked tokens in the context of MGM. We provide more details about the derivation in Appendix B.3. In this study, we select MaskGCT (Wang et al., 2025a) as a representative to apply this proposed algorithm for its text-to-codec stage.

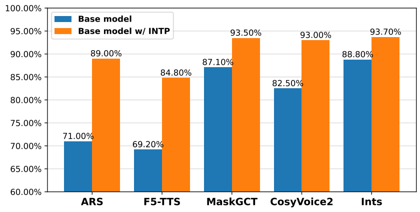
5 Experiments

Evaluation Data We evaluate zero-shot TTS systems across diverse domains in both English and Chinese languages. Based on SeedTTS's evaluation samples (Anastassiou et al., 2024) (which are widely used and also serve as the evaluation set for the pre-trained models of ARS (Wang et al., 2025a), F5-TTS (Chen et al., 2024c), MaskGCT (Wang et al., 2025a), and CosyVoice 2 (Du et al., 2024b) in this study), we construct evaluation sets across four distinct domains: (1) **Regular cases:** We use SeedTTS test-en (1,000 samples) and SeedTTS test-zh datasets (2,000 samples). (2) **Articulatory cases:** These involve tongue twisters and repeated texts. For Chinese, we use SeedTTS test-hard, while for English, we use reference speech prompts of SeedTTS test-en, and employ Deepseek-V3 (DeepSeek-AI et al., 2024) to construct the articulatory texts like SeedTTS test-hard. There are 800 samples in total. (3) **Code-switching cases:** These target texts are a mixture of English and Chinese. Based on SeedTTS test-en and test-zh, we keep their reference speech prompts unchanged, and adopt Deepseek-V3 to transform their texts into code-switching style. There are 1,000 samples in total. (4) **Cross-lingual cases:** We construct two types of cross-lingual samples: *zh2en* (500 samples) and *en2zh* (500 samples). The zh2en means Chinese reference speech (from SeedTTS test-zh) with English target text (from SeedTTS test-en). Similarly for en2zh. The detailed distribution of these sets is presented in Table 11, Appendix F.1.

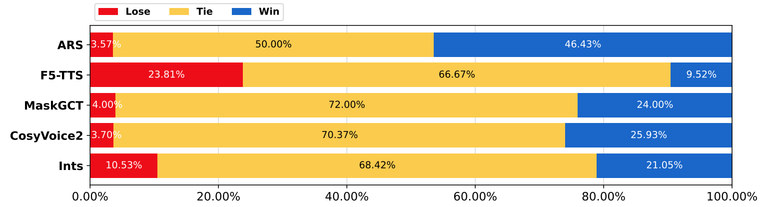
Evaluation Metrics For objective metrics, we evaluate the intelligibility (WER, ↓), speaker similarity (SIM, ↑), and overall speech quality (UTMOS (Saeki et al., 2022), ↑). Specifically, for WER, we employ `Whisper-large-v3` (Radford et al., 2023) for English, and `Paraformer-zh` (Gao et al., 2022, 2023) for Chinese and code-switching texts. For SIM, we compute the cosine similarity between the WavLM TDNN (Chen et al., 2022) speaker embeddings of generated samples and the reference speeches. For subjective metrics, we employ Comparative Mean Opinion Score (rated from -2 to 2) to evaluate naturalness (N-CMOS, ↑), use reading accuracy (Section 3.4) to evaluate intelligibility, and use A/B Testing to compare speaker similarity between the generated samples before and after intelligibility alignment. Detailed descriptions of all the metrics are provided in Appendix F.

5.1 Effect of DPO with INTP

To verify the effectiveness of DPO with INTP for existing TTS models, we conduct alignment experiments with multiple models. In addition to ARS, F5-TTS, and MaskGCT, which were used in constructing the INTP dataset, we also introduce two more powerful models in terms of intelligi-



(a) Comparison of reading accuracy.



(b) Win/Lose/Tie of speaker similarity after INTP alignment.

Figure 2: Subjective evaluation of intelligibility and speaker similarity for models before and after INTP alignment.

bility: CosyVoice 2 (Du et al., 2024b) and Ints (Appendix C), to validate INTP's weak-to-strong generalization capability. The experimental results are presented in Table 4, including results on the objective WER, SIM, and the subjective naturalness CMOS.

We observe three key findings from Table 4: (1) Across different evaluation cases, while almost all models demonstrate strong intelligibility performance in regular cases (WER < 4.0), they struggle significantly with articulatory, code-switching, and cross-lingual cases. We show some hallucinated outputs for these domains on our demo website. (2) Comparing across models, CosyVoice 2 and Ints achieves better average intelligibility performance across all cases (WER of 13.09 and 11.97), highlighting the strength of using a textual LLM as the initialization of large-scale TTS model (Du et al., 2024b). (3) Through DPO with INTP, all models, including the more intelligible CosyVoice 2 and Ints that are out of the INTP distribution, show improvements in both intelligibility (WER) and naturalness (N-CMOS), and display comparable performance for speaker similarity (SIM).

Furthermore, we randomly sample 300 samples for subjective evaluation, including assessments of reading accuracy and A/B testing of speaker similarity before and after INTP alignment (see Appendix F.3 for details). The results in Figure 2 demonstrate that INTP alignment enhances all five models in terms of both intelligibility (higher reading accuracy in Figure 2a) and speaker similarity (more Tie/Win percentages in Figure 2b).

5.2 Effect of Different Data within INTP

To investigate the impact of different distributions within INTP, we conduct ablation studies from multiple perspectives. In Table 5, we present three groups of experiments on ARS: the effect of data across different text types, across different models, and the effect of different negative samples. Additional results, including the effect of data across different languages are provided in Appendix E.

We observe three key findings from Table 5: (1) Group 1 demonstrates that different scenarios require customized post-training data. For instance, repeated data proves particularly effective for articulatory cases, while pronunciation-perturbed data significantly improves pronunciation accuracy and WER in cross-lingual cases (see our demo website for details). Moreover, utilizing data from multiple scenarios (i.e., the complete INTP) yields the best overall improvements. (2) Group 2 reveals that model improvement can be achieved through alignment using synthetic data, regardless of whether it's generated by the model itself or other models. Besides, the intra-pairs and inter-pairs are complementary for model improvements. (3) Group 3 shows that using only positive samples from INTP for supervised fine-tuning (SFT) can already improve quality. Building upon this, incorporating negative samples for preference learning leads to even more substantial gains.

5.3 Iterative Intelligibility Alignment

Furthermore, we explore how to establish an *iterative* preference alignment, i.e., data and model flywheel (Bai et al., 2022; Dubey et al., 2024; Xiong et al., 2024). This approach aligns with the online reinforcement learning (RL) framework Li et al. (2024b). We investigate two rounds of alignment based on Ints, where Ints v1 (INTP-aligned model) is used to generate new preference data for training Ints v2, following a similar *cadence* of data collection as (Bai et al., 2022). To prepare Ints v1 generated preference data, we sample a challenging prompt subset from INTP and adopt the same pipeline as INTP to construct preference pairs (see Appendix C.2 for details). The results of this iterative alignment are shown in Table 6. We can observe that compared to Ints v1, Ints v2 yields additional improvements across all scenarios, which demonstrates that effectiveness of iterative alignment. However, we observe that the magnitude of improvement in the second round is notably smaller than the first round. We suspect this indicates that

Model	Regular cases			Articulatory cases			Code-switching cases			Cross-lingual cases			Avg		
	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS
Group 1: Effect of Data across Different Text Types															
ARS (Wang et al., 2025a)	3.96	0.717	3.145	20.03	0.693	2.915	54.15	0.693	3.045	19.76	0.630	3.120	24.47	0.683	3.056
w/ Regular	2.45	0.727	3.200	17.41	0.706	3.000	37.52	**0.701**	3.110	9.66	**0.638**	3.200	16.76	**0.693**	3.128
w/ Repeated	2.33	0.725	3.225	12.88	0.711	3.050	39.74	**0.701**	3.150	10.96	0.636	3.235	16.48	**0.693**	3.165
w/ Code-switching	2.32	**0.729**	3.220	17.67	0.704	3.050	**34.20**	0.695	3.140	8.69	0.633	3.215	15.72	0.690	3.156
w/ Pronunciation-perturbed	**2.21**	0.720	**3.250**	17.76	0.693	**3.075**	35.99	0.687	3.185	**8.24**	0.617	**3.285**	16.05	0.679	**3.199**
w/ Punctuation-perturbed	2.46	0.722	3.240	17.35	0.699	3.020	42.73	0.694	3.160	10.94	0.624	3.255	18.37	0.684	3.169
w/ INTP	2.32	0.727	3.210	**12.83**	**0.713**	3.035	36.91	0.698	3.145	9.57	0.632	3.250	**15.41**	0.692	3.160
Group 2: Effect of Data across Different Models															
ARS (Wang et al., 2025a)	3.96	0.717	3.145	20.03	0.693	2.915	54.15	0.693	3.045	19.76	0.630	3.120	24.47	0.683	3.056
w/ ARS pairs	2.56	0.717	3.200	13.05	0.705	3.015	40.91	0.691	3.125	11.07	0.622	3.225	16.90	0.684	3.141
w/ MaskGCT pairs	2.37	0.724	**3.210**	16.85	0.700	3.010	37.41	0.692	3.105	**8.83**	0.625	3.200	16.37	0.685	3.131
w/ F5-TTS pairs	2.46	0.721	**3.210**	14.99	0.705	**3.035**	38.77	0.690	3.115	10.01	0.621	3.225	16.56	0.684	3.146
w/ Intra pairs	2.33	0.721	3.200	15.29	0.705	3.015	37.99	0.687	3.115	9.36	0.624	3.200	16.24	0.684	3.133
w/ Inter pairs	**2.25**	0.726	3.180	15.42	0.703	2.965	38.69	0.697	3.065	10.61	0.631	3.170	16.74	0.689	3.095
w/ INTP	2.32	**0.727**	**3.210**	**12.83**	**0.713**	**3.035**	**36.91**	**0.698**	**3.145**	9.57	**0.632**	**3.250**	**15.41**	**0.692**	**3.160**
Group 3: Effect of Different Negative Samples															
ARS (Wang et al., 2025a)	3.96	0.717	3.145	20.03	0.693	2.915	54.15	0.693	3.045	19.76	0.630	3.120	24.47	0.683	3.056
w/ Regular (SFT)*	3.28	0.716	3.165	20.03	0.685	2.935	48.73	0.691	3.065	17.25	0.630	3.165	22.32	0.680	3.083
w/ Regular*	2.45	**0.727**	3.200	17.41	**0.706**	3.000	37.52	**0.701**	3.110	9.66	**0.638**	3.200	16.76	**0.693**	3.128
w/ Pronunciation-perturbed*	**2.21**	0.720	**3.250**	17.76	0.693	**3.075**	**35.99**	0.687	**3.185**	**8.24**	0.617	**3.285**	**16.05**	0.679	**3.199**
w/ Punctuation-perturbed*	2.46	0.722	3.240	**17.35**	0.699	3.020	42.73	0.694	3.160	10.94	0.624	3.255	18.37	0.684	3.169

* The positive samples in these four experiments are identical. **w/ Regular (SFT)** refers to supervised fine-tuning using positive samples only, excluding negative samples. **w/ Regular** employs WER-based negative samples, while the other two utilize our proposed human-guided negative samples.

Table 5: Effect of different data within INTP for ARS.

Model	Preference Data	Regular cases			Articulatory cases			Code-switching cases			Cross-lingual cases			Avg		
		WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS
Ints	-	3.14	0.688	3.175	12.08	0.666	3.025	22.88	0.646	3.045	9.78	0.572	3.150	11.97	0.643	3.099
Ints v1	**INTP**	2.36	0.686	3.205	9.38	0.664	3.060	13.80	0.642	3.125	6.28	0.571	3.230	7.96	0.641	3.155
Ints v2	**Ints v1 generated**	2.21	0.686	3.210	8.48	0.660	3.085	12.33	0.643	3.140	5.40	0.567	3.250	7.10	0.639	3.171

Table 6: Iterative Preference Alignment for Ints.

the upper bound of iterative alignment is largely determined by the base model's inherent capabilities, suggesting future research should focus on base models with higher potential.

6 Conclusion

In this work, we focus on the intelligibility issues of modern zero-shot TTS systems across diverse domains, especially in hard-to-pronounce texts, code-switching, and cross-lingual synthesis. We propose to address these challenges using preference alignment with our newly constructed INTP dataset, which contains diverse preference pairs determined through model self-comparison, cross-model comparison, and human guidance. We employ DPO and design special extensions to significantly improve various TTS architectures, while demonstrating INTP's weak-to-strong generalization capability and establishing an iterative preference alignment flywheel with more powerful base models.

Limitations

While our approach demonstrates significant improvements in zero-shot TTS intelligibility across diverse domains, several limitations remain. Although INTP covers multiple challenging scenarios, it may not fully capture all edge cases, such as specialized jargon or rare language pairs. Future work could expand to more low-resource languages and niche domains. Besides, constructing INTP and conduct alignment experiments on large models like Ints require substantial computational resources, potentially limiting accessibility.

Potential Risks

The proposed method introduces several risks that warrant consideration. Enhanced TTS systems could be exploited to generate deceptive content (e.g., deepfake audio), posing ethical challenges. Robust safeguards and watermarking mechanisms are critical for deployment. While INTP uses public datasets, real-world applications may risk incorporating sensitive or copyrighted speech data, requiring strict governance protocols.

Acknowledgment

This work is partially supported by the NSFC under Grant 62376237, Shenzhen Science and Technology Program ZDSYS20230626091302006, and Shenzhen Research Institute of Big Data (Internal Project Fund, Grant No. T00120230002). We appreciate Yushun Zhang and the anonymous reviewers for their insightful comments and suggestions.

References

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint*, abs/2404.14219.

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, Zhen Wei, Jian Wu, Chao Yao, Yifeng Yang, Yuanhao Yi, Junteng Zhang, Qidi Zhang, Shuo Zhang, Wenjie Zhang, Yang Zhang, Zilin Zhao, Dejian Zhong, and Xiaobin Zhuang. 2024. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint*, abs/2406.02430.

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint*, abs/2204.05862.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matthew Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2023a. Audiolm: A language modeling approach to audio generation. *IEEE ACM Trans. Audio Speech Lang. Process.*, 31:2523–2533.

Zalán Borsos, Matthew Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. 2023b. Soundstorm: Efficient parallel audio generation. *arXiv preprint*, abs/2305.09636.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. 2024. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *ICML*. OpenReview.net.

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

Jingyi Chen, Ju-Seung Byun, Micha Elsner, and Andrew Perrault. 2024b. Dlpo: Diffusion model loss-guided reinforcement learning for fine-tuning text-to-speech diffusion models. *arXiv preprint*, abs/2405.14632.

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):1505–1518.

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

Geoffrey Cideron, Sertan Girgin, Mauro Verzetti, Damien Vincent, Matej Kastelic, Zalán Borsos, Brian McWilliams, Victor Ungureanu, Olivier Bachem, Olivier Pietquin, Matthieu Geist, Léonard Hussenot, Neil Zeghidour, and Andrea Agostinelli. 2024. Musicrl: Aligning music generation to human preferences. In *ICML*. OpenReview.net.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. Deepseek-v3 technical report. *arXiv preprint*, abs/2412.19437.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2023. High fidelity neural audio compression. *Trans. Mach. Learn. Res.*, 2023.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, Zhifu Gao, and Zhijie Yan. 2024a. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint*, abs/2407.05407.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, Fan Yu, Huadai Liu, Zhengyan Sheng, Yue Gu, Chong Deng, Wen Wang, Shiliang Zhang, Zhijie Yan, and Jingren Zhou. 2024b. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint*, abs/2412.10117.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *arXiv preprint*, abs/2407.21783.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, Yanqing Liu, Sheng Zhao, and Naoyuki Kanda. 2024. E2 TTS: embarrassingly easy fully non-autoregressive zero-shot TTS. In *SLT*. IEEE.

Deqing Fu, Tong Xiao, Rui Wang, Wang Zhu, Pengchuan Zhang, Guan Pang, Robin Jia, and Lawrence Chen. 2024. TLDR: token-level detective reward model for large vision language models. *arXiv preprint*, abs/2410.04734.

Xiaoxue Gao, Chen Zhang, Yiming Chen, Huayun Zhang, and Nancy F. Chen. 2024. Emo-dpo: Controllable emotional speech synthesis through direct preference optimization. *arXiv preprint*, abs/2409.10157.

Zhifu Gao, Zerui Li, Jiaming Wang, Haoneng Luo, Xian Shi, Mengzhe Chen, Yabin Li, Lingyun Zuo, Zhihao Du, and Shiliang Zhang. 2023. Funasr: A fundamental end-to-end speech recognition toolkit. In *INTERSPEECH*, pages 1593–1597. ISCA.

Zhifu Gao, Shiliang Zhang, Ian McLoughlin, and Zhijie Yan. 2022. Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition. In *INTERSPEECH*, pages 2063–2067. ISCA.

Hao-Han Guo, Kun Liu, Fei-Yu Shen, Yi-Chen Wu, Feng-Long Xie, Kun Xie, and Kai-Tuo Xu. 2024. Fireredtts: A foundation text-to-speech framework for industry-level generative speech applications. *arXiv preprint*, abs/2409.03283.

Tingwei Guo, Cheng Wen, Dongwei Jiang, Ne Luo, Ruixiong Zhang, Shuaijiang Zhao, Wubo Li, Cheng Gong, Wei Zou, Kun Han, et al. 2021. 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.

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. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *SLT*. IEEE.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al. 2025. Emilia: A large-scale, extensive, multilingual, and diverse dataset for speech generation. *arXiv preprint*, 2501.15907.

Yuchen Hu, Chen Chen, Siyin Wang, Eng Siong Chng, and Chao Zhang. 2024. Robust zero-shot text-to-speech synthesis with reverse inference optimization. *arXiv preprint*, abs/2407.02243.

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

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Eric Liu, Yichong Leng, Kaitao Song, Siliang Tang, et al. 2024. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *Forty-first International Conference on Machine Learning*.

Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. IEEE.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. In *NeurIPS*.

Jiaqi Li, Xiaolong Lin, Zhekai Li, Shixi Huang, Yuancheng Wang, Chaoren Wang, Zhenpeng Zhan, and Zhizheng Wu. 2025a. Dualcodec: A low-frame-rate, semantically-enhanced neural audio codec for speech generation. In *INTERSPEECH*. ISCA.

Jiaqi Li, Xueyao Zhang, Yuancheng Wang, Haorui He, Chaoren Wang, Li Wang, Huan Liao, Junyi Ao, Zeyu Xie, Yiqiao Huang, Junan Zhang, and Zhizheng Wu. 2025b. Overview of the amphion toolkit (v0.2). *arXiv preprint arXiv:2501.15442*.

Ziniu Li, Tian Xu, and Yang Yu. 2024a. When is rl better than dpo in rlhf? a representation and optimization perspective. In *The Second Tiny Papers Track at ICLR 2024*.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2024b. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *Forty-first International Conference on Machine Learning*.

Huan Liao, Haonan Han, Kai Yang, Tianjiao Du, Rui Yang, Qinmei Xu, Zunnan Xu, Jingquan Liu, Jiasheng Lu, and Xiu Li. 2024. BATON: aligning text-to-audio model using human preference feedback. In *IJCAI*, pages 4542–4550. ijcai.org.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. Flow matching for generative modeling. In *ICLR*. OpenReview.net.

Navonil Majumder, Chia-Yu Hung, Deepanway Ghosal, Wei-Ning Hsu, Rada Mihalcea, and Soujanya Poria. 2024. Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization. In *ACM Multimedia*, pages 564–572. ACM.

Paarth Neekhara, Shehzeen Hussain, Subhankar Ghosh, Jason Li, Rafael Valle, Rohan Badlani, and Boris Ginsburg. 2024. Improving robustness of llm-based speech synthesis by learning monotonic alignment. In *INTERSPEECH*. ISCA.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

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

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. UTMOS: utokyo-sarulab system for voicemos challenge 2022. In *INTERSPEECH*, pages 4521–4525. ISCA.

Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. A comprehensive survey of hallucination in large language, image, video and audio foundation models. In *EMNLP (Findings)*, pages 11709–11724. Association for Computational Linguistics.

Kai Shen, Zeqian Ju, Xu Tan, Eric Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. 2024. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *ICLR*. OpenReview.net.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. In *NeurIPS*, volume 35, pages 9460–9471.

Xu Tan. 2023. *Neural Text-to-Speech Synthesis*. Springer.

Jinchuan Tian, Chunlei Zhang, Jiatong Shi, Hao Zhang, Jianwei Yu, Shinji Watanabe, and Dong Yu. 2024. Preference alignment improves language model-based TTS. *arXiv preprint*, abs/2409.12403.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. 2024. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint*, abs/2301.02111.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2025a. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. In *ICLR*. OpenReview.net.

Yuancheng Wang, Jiachen Zheng, Junan Zhang, Xueyao Zhang, Huan Liao, and Zhizheng Wu. 2025b. Metis: A foundation speech generation model with masked generative pre-training. *arXiv preprint arXiv:2502.03128*.

Lilian Weng. 2024. Reward hacking in reinforcement learning. *lilianweng.github.io*.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. 2024. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under kl-constraint. In *ICML*. OpenReview.net.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. 2023. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *NeurIPS*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint*, abs/2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *arXiv preprint*, abs/2412.15115.

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*, abs/2502.02950.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Dong Zhang, Zhaowei Li, Shimin Li, Xin Zhang, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2024a. Speechalign: Aligning speech generation to human preferences. In *NeurIPS*.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024b. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.

Xueyao Zhang, Liumeng Xue, Yicheng Gu, Yuancheng Wang, Jiaqi Li, Haorui He, Chaoren Wang, Ting Song, Xi Chen, Zihao Fang, Haopeng Chen, Junan Zhang, Tze Ying Tang, Lexiao Zou, Mingxuan Wang, Jun Han, Kai Chen, Haizhou Li, and Zhizheng Wu. 2024c. Amphion: An open-source audio, music and speech generation toolkit. In *IEEE Spoken Language Technology Workshop, SLT 2024*.

Xueyao Zhang, Xiaohui Zhang, Kainan Peng, Zhenyu Tang, Vimal Manohar, Yingru Liu, Jeff Hwang, Dangna Li, Yuhao Wang, Julian Chan, Yuan Huang, Zhizheng Wu, and Mingbo Ma. 2025. Vevo: Controllable zero-shot voice imitation with self-supervised disentanglement. In *ICLR*. OpenReview.net.

Yu Zhang, Changhao Pan, Wenxiang Guo, Ruiqi Li, Zhiyuan Zhu, Jialei Wang, Wenhao Xu, Jingyu Lu, Zhiqing Hong, Chuxin Wang, Lichao Zhang, Jinzheng He, Ziyue Jiang, Yuxin Chen, Chen Yang, Jiecheng Zhou, Xinyu Cheng, and Zhou Zhao. 2024d. Gtsinger: A global multi-technique singing corpus with realistic music scores for all singing tasks. In *NeurIPS*.

A Construction Details of INTP

A.1 Prompt Construction

We construct English and Chinese prompt data, both based on the Emilia-Large dataset (He et al., 2024, 2025), which contains diverse real-world speech data across various topics, recording scenarios, and speaking styles.

Reference Speech We perform stratified sampling on Emilia-Large's speech data based on its metadata such as topics and tags to cover diverse acoustic conditions. Considering the memory constraints of existing zero-shot TTS models during inference, we only select samples with durations not exceeding 12 seconds.

Target Text Similarly to reference speech, we perform stratified sampling based on Emilia-Large's metadata to cover diverse semantic topics. We select speech samples with durations between 5 and 22 seconds, and use their corresponding textual transcriptions as the target text data source.

We utilize DeepSeek V3 (DeepSeek-AI et al., 2025) to preprocess the sampled textual transcriptions, such as typo correction and punctuation mark normalization, and use the processed text as regular text in INTP. Specifically, we use the following instruction for DeepSeek V3 to conduct text preprocessing:

System Prompt:
I obtained a text from an audio file based on some ASR models. Please help me clean it up (e.g., correct typos, add proper punctuation marks, and make the sentences semantically coherent). Note: (1) You can modify, add, or replace words that better fit the context to ensure semantic coherence. (2) Please only return the cleaned-up result without any explanation.

User Prompt (Example):
a panda eats shoes and leaves

System Output (Example):
A panda eats shoots and leaves.

Furthermore, we employ DeepSeek V3 to transform the regular text into different types. To generate Chinese-English-mixed code-switching texts:

System Prompt:
请你把这句话，转换成一个中文、英文混合的 code-switching 版本。注意：你只需要返回给我转换后的结果，不需要任何解释。

User Prompt (Example):
A panda eats shoots and leaves.

System Output (Example):

熊猫吃 shoots 和 leaves。

To generate punctuation-perturbed texts:

System Prompt:
假设你是一个 Text To Speech (TTS) 领域的专家，现在，让我们对一个 TTS 系统进行攻击。具体地：我输入一个文本，请你修改这条文本里面的若干词语，从而使 TTS 系统更容易出错。例如：你可以修改为把某些字修改为容易读错的形近字、把多音字做替换，等等，但你不要增加和删除原有的文本。注意：你只需要返回给我转换后的结果，不需要任何解释。

例子1:
【我的输入】我今天很高兴
【你的输出】窝锦添狠搞醒

例子2:
【我的输入】目前，爱心人士正在种作寄养的小猫已经五个月大了。而本人的种作寄养申请单需要进一步审核。为了避免小猫多次转手，治疗者们对小猫的种作寄养提出了严格要求：申请人需年满二十三岁。
【你的输出】幕前，爱信人士正在重作寄扬的削猫已经伍个月大了。而本人的重作寄扬神情但需要进一步审核。为了闭面削猫多次转售，治理者们对削猫的重作寄扬提出了阉割要求：申情人需年慢贰拾叁岁。

例子3:
【我的输入】And the idea of standing all by himself in a crowded market, to be pushed and hired by some big, strange farmer, was very disagreeable. Why not sing that high note and grow potatoes?
【你的输出】And the eye dear of standing awl bye himself in a crowd dead market, two bee pushed and high red buy sum big, strange far mer, was vary dis agreeable. Y knot sing that hi note and grow poe eight toes?

User Prompt (Example):
A panda eats shoots and leaves.

System Output (Example):
A pan duh eights shots n leafs.

To generate repeated text and punctuation-perturbed text, we leverage DeepSeek V3 to create executable Python scripts that implement rule-based word repetition and random punctuation modification. These scripts will be included in our future open-source repository.

Combination between Speech and Text Based on the language of reference speech and target text data, we design four balanced combination categories: monolingual combinations (*en2en* and *zh2zh*) and cross-lingual combinations (*zh2en* and *en2zh*), where *zh2en* denotes Chinese reference speech with English target text, and similarly for others. For each text type shown in Table 1a (Regular, Repeated, Code-Switching, Pronunciation-perturbed, and Punctuation-perturbed), we construct 12K prompts.

A.2 Model Selection

- **ARS** (Wang et al., 2025a): We use the original checkpoint (pre-trained on Emilia) provided by the authors.
- **F5-TTS** (Chen et al., 2024c): We use the officially released checkpoint[2] for INTP data generation.
- **MaskGCT** (Wang et al., 2025a): We use the officially released checkpoint[3] (Zhang et al., 2024c; Li et al., 2025b) for INTP data generation.

In addition to these three models used for INTP construction, we also investigate INTP's effectiveness on **CosyVoice 2** and **Ints**. For CosyVoice 2, we conduct alignment experiments using its officially released checkpoint[4] as the base model. Details of the pre-trained models of Ints are provided in Appendix C.

A.3 Preference Pairs Construction

A.3.1 Intra Pair

For each model and prompt, we perform five samplings and construct intra pairs based on their WER comparisons. To maximize the performance gap between positive and negative samples, we employ two strategies. First, we use diverse hyperparameters during the five generations to increase sample diversity, selecting the generation with the lowest WER as positive samples and the highest WER as negative samples. Second, we apply a threshold to filter out pairs where the WER gap between positive and negative samples is less than 6.0.

Specifically, for ARS's five samplings, we set top k to 20 and top p to 1.0, while using different temperatures of 0.4, 0.6, 0.8, 1.0, and 1.2. For F5-TTS and MaskGCT, we use the generated speech target duration as the sampling hyperparameter. Denoting the "ground truth" duration[5] as d, we employ five different duration parameters: $0.8d$, $0.9d$, $1.0d$, $1.1d$, and $1.2d$.

[2] https://huggingface.co/SWivid/F5-TTS
[3] https://huggingface.co/amphion/MaskGCT
[4] https://github.com/FunAudioLLM/CosyVoice
[5] Since we use Emilia-Large's transcription data as target text in our prompt construction process (Appendix A.1), we refer to the original speech duration corresponding to this transcription as the "ground truth" duration.

A+2	A+1	Tie	B+1	B+2
10.9%	29.0%	15.0%	32.4%	12.6%

* For each pair, we present the two samples to human raters in random order, labeled as A and B. A+2 indicates that sample A's naturalness is significantly better than B, while A+1 indicates that sample A is moderately better than B, similar for B+2 and B+1. Tie indicates no perceptible difference.

Table 7: Human naturalness preference for 1,000 pairs from INTP regular text domain.

	Naturalness Winner	Naturalness Tie	Naturalness Loser
INTP winner	72%	15%	13%

Table 8: Agreement between INTP preference and human naturalness preference.

A.3.2 Inter Pair

We construct inter pairs based on the intra pairs established in Appendix A.3.1. For a given prompt, we denote model A's intra pair as (y_A^w, y_A^l) and model B's intra pair as (y_B^w, y_B^l). We construct inter pairs through three types of comparisons: between y_A^w and y_B^w, between y_A^w and y_B^l, and between y_A^l and y_B^w. Note that we exclude comparisons between y_A^l and y_B^l to ensure high quality of positive samples. We apply the same WER threshold as in Appendix A.3.1 to filter out pairs with small performance gaps.

A.3.3 Perturbed Pair

The instructions used to prompt DeepSeek V3 for obtaining pronunciation-perturbed and punctuation-perturbed texts are shown in Appendix A.1. Specifically, we only use data from INTP's regular text domain to construct perturbed pairs.

A.4 Human Verification

In Section 3.4, we evaluated INTP's alignment with human intelligibility perception. In this section, we investigate the alignment between INTP and human naturalness preferences. Specifically, we design a **naturalness preference** annotation task (Appendix F.3). We randomly sample 1,000 pairs from INTP's regular text domain for human annotation, with results shown in Table 7 and 8. The results reveal two key findings: First, 85% of INTP pairs exhibit distinguishable naturalness preferences (Tie rate of 15% in Table 7). Additionally, INTP's preference determination shows strong agreement with human naturalness preferences (72% agreement rate between INTP winners and naturalness winners in Table 8). These results suggest that INTP can also serve as a foundation dataset for naturalness preference alignment in future research.

B Details of the Derivation

B.1 DPO for AR Models

Starting from Equation 2, Rafailov et al. (2023) demonstrate that the optimization problem admits a closed-form solution. Specifically, the optimal policy $p_\theta^*(y|x)$ that maximizes the RL objective is given by:

$$p_\theta^*(y|x) = \frac{1}{Z(x)} p_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right), \tag{10}$$

where $Z(x)$ is the partition function ensuring normalization. This establishes a direct relationship between the reward function and the policy:

$$r(x,y) = \beta \log \frac{p_\theta^*(y|x)}{p_{\text{ref}}(y|x)} + \beta \log Z(x). \tag{11}$$

Substituting this reward expression (Equation 11) into the reward modeling loss function (Equation 1) leads the DPO loss (Equation 3), which we represent here as:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{\mathcal{D}}\left[\log \sigma\left(\beta\left(\log \frac{p_\theta(y_w|x)}{p_{\text{ref}}(y_w|x)} - \log \frac{p_\theta(y_l|x)}{p_{\text{ref}}(y_l|x)}\right)\right)\right].$$

B.2 DPO for Flow-Matching Models

Starting from Equation 5, which we represent here as:

$$\max_{p_\theta} \mathbb{E}_{y_1 \sim p_\theta(y_1|x), t, x}[r(y_1, x)] - \beta \mathbb{D}_{\text{KL}}[p_\theta(y_1|y_t,t,x) \| p_{\text{ref}}(y_1|y_t,t,x)].$$

Similar to the derivation in DPO (Rafailov et al., 2023) and Wallace et al. (2024), we obtain the closed-form solution for the optimal policy as:

$$p_\theta^*(y_1|y_t,t,x) = \frac{1}{Z(y_t,t,x)} p_{\text{ref}}(y_1|y_t,t,x) \exp\left(\frac{1}{\beta} r(y_1,x)\right), \tag{12}$$

where $Z(y_t,t,x)$ is the partition function ensuring normalization. We can then express the reward model $r(y_1,x)$ as:

$$r(y_1,x) = \beta \log \frac{p_\theta^*(y_1|y_t,t,x)}{p_{\text{ref}}(y_1|y_t,t,x)} + \beta \log Z(y_t,t,x). \tag{13}$$

Similarly, substituting this reward expression (Equation 13) into the reward modeling loss function (Equation 1) leads to the DPO loss for OT-FM (Equation 6), which we represent here:

$$\mathcal{L}_{\text{DPO-FM}} = -\mathbb{E}_{(y_1^w, y_1^l, x) \sim \mathcal{D}, t} \log \sigma\left(\beta\left(\log \frac{p_\theta(y_1^w|y_t^w,t,x)}{p_{\text{ref}}(y_1^w|y_t^w,t,x)} - \log \frac{p_\theta(y_1^l|y_t^l,t,x)}{p_{\text{ref}}(y_1^l|y_t^l,t,x)}\right)\right).$$

Reviewing the training objective of OT-FM (Equation 4), we find that it is equivalent to fitting a Gaussian likelihood. In other words, the induced likelihood can be interpreted as:

$$p_\theta(y_1 \mid y_t, t, x) \propto \exp\left(-\frac{1}{\beta}\left\|v_\theta(y_t, t, x) - (y_1 - y_0)\right\|_2^2\right),$$

similarly, for the reference policy, we have:

$$p_{\text{ref}}(y_1 \mid y_t, t, x) \propto \exp\left(-\frac{1}{\beta}\left\|v_{\text{ref}}(y_t, t, x) - (y_1 - y_0)\right\|_2^2\right).$$

Here, β serves as an inverse temperature (or noise variance), and the normalization constants cancel out when taking the ratio. By taking the logarithm of the ratio between the learned policy and the reference policy, we obtain:

$$\log \frac{p_\theta(y_1 \mid y_t, t, x)}{p_{\text{ref}}(y_1 \mid y_t, t, x)} = -\frac{1}{\beta}\Big(\|v_\theta(y_t, t, x) - (y_1 - y_0)\|_2^2 - \|v_{\text{ref}}(y_t, t, x) - (y_1 - y_0)\|_2^2\Big).$$

Multiplying both sides by β results in:

$$\beta \log \frac{p_\theta(y_1 \mid y_t, t, x)}{p_{\text{ref}}(y_1 \mid y_t, t, x)} = -\Big(\|v_\theta(y_t, t, x) - (y_1 - y_0)\|_2^2 - \|v_{\text{ref}}(y_t, t, x) - (y_1 - y_0)\|_2^2\Big).$$

By substituting the log-ratio formulation into Equation 6, we can transform the DPO loss for OT-FM into a form related to the velocity, as shown in Equation 7, which is represented as:

$$\mathcal{L}_{\text{DPO-FM}} = -\mathbb{E}_{(y_1^w, y_1^l, x)\sim\mathcal{D}, t} \log \sigma\Big(-\beta \Big(\|v_\theta(y_t^w, t, x) - (y_1^w - y_0^w)\|_2^2 - \|v_{\text{ref}}(y_t^w, t, x) - (y_1^w - y_0^w)\|_2^2\Big) - \Big(\left\|v_\theta(y_t^l, t, x) - (y_1^l - y_0^l)\right\|_2^2 - \left\|v_{\text{ref}}(y_t^l, t, x) - (y_1^l - y_0^l)\right\|_2^2\Big)\Big).$$

B.3 DPO for Masked Generative Models

Similar to flow-matching, let $p_\theta(y_0 \mid y_t, x)$ denote the policy to be optimized, and $p_{\text{ref}}(y_0 \mid y_t, x)$ the reference policy. We can rewrite the RL objective for MGM as follows:

$$\max_{p_\theta} \; \mathbb{E}_{y_0 \sim p_\theta(y_0|x), t, x}\left[r(y_0, x)\right] - \beta \mathbb{D}_{\text{KL}}\left[p_\theta(y_0|y_t, x) \,\|\, p_{\text{ref}}(y_0|y_t, x)\right]. \tag{14}$$

We can also derive the closed-form solution for the optimal policy:

$$p_\theta^*(y_0|y_t, x) = \frac{1}{Z(y_t, x)} p_{\text{ref}}(y_0|y_t, x) \exp\left(\frac{1}{\beta} r(y_0, x)\right), \tag{15}$$

and express the reward model as follows:

$$r(y_0, x) = \beta \log \frac{p_\theta^*(y_0|y_t, x)}{p_{\text{ref}}(y_0|y_t, x)} + \beta \log Z(y_t, x), \tag{16}$$

where $Z(y_t, x)$ is the partition function ensuring normalization. Also, substituting this reward expression (Equation 16) into the reward modeling loss function (Equation 1) leads to the DPO loss for MGM:

$$\mathcal{L}_{\text{DPO-MGM}} = -\mathbb{E}_{(y^w, y^l, x)\sim\mathcal{D}, t} \log \sigma\left(\beta\left(\log \frac{p_\theta(y_0^w|y_t^w, x)}{p_{\text{ref}}(y_0^w|y_t^w, x)} - \log \frac{p_\theta(y_0^l|y_t^l, x)}{p_{\text{ref}}(y_0^l|y_t^l, x)}\right)\right). \tag{17}$$

Here, y_t^w and y_t^l are masked versions of y_0^w and y_0^l generated via the mask schedule $\gamma(t)$. Note that $p_\theta(y_0|y_t, x)$ corresponds to the sum of the log-probabilities of the unmasked tokens in the context of MGM.

C Ints: Intelligibility-enhanced Speech Language Model

Ints is an intelligibility-enhanced speech language model. It follows a two-stage generation paradigm like (Anastassiou et al., 2024; Du et al., 2024a; Wang et al., 2025a): in the first stage, it uses an AR model to generate discrete speech tokens, while in the second stage, it employs a flow matching model to generate mel-spectrograms from speech tokens. We use the first-layer tokens from DualCodec (Li et al., 2025a) as the modeling target for the first stage of Ints, due to its efficient compression representation (12.5Hz tokens for 24kHz speech) and rich semantic information. Particularly, the first-stage AR model is directly **initialized from a large language model** while extending the vocabulary to include speech tokens. The codebook size of speech tokens is 16,384. Specifically, in this work, we use the 3.8B `Phi-3.5-mini-instruct`[6] (Abdin et al., 2024), motivated by scaling up model size and leveraging the rich textual semantic knowledge.

C.1 TTS Instruction Design

We format the input as a text-to-speech instruction concatenated with speech tokens. The input sequence is represented as:

$$[\boldsymbol{I}, \boldsymbol{T}, <|\text{startofspeech}|>, \boldsymbol{S}, <|\text{endofspeech}|>]$$

where $\boldsymbol{I}$ is the instruction prefix (e.g., "Please speak the following text out loud"), and $\boldsymbol{T}$ and $\boldsymbol{S}$ denote the text and speech token sequences, respectively. The special tokens $<|\text{startofspeech}|>$ and $<|\text{endofspeech}|>$ mark the boundaries of the speech token sequence.

[6] https://huggingface.co/microsoft/Phi-3.5-mini-instruct

Model	Regular (*en*)			Articulatory (*en*)			Code-switching (*en2mixed*)			Cross-lingual (*zh2en*)			Avg		
On English Evaluation Samples															
	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS
ARS (Wang et al., 2025a)	3.55	0.682	3.560	15.98	0.675	3.400	48.59	0.629	3.190	15.22	0.697	3.150	20.84	0.671	3.325
w/ en2en	**1.96**	**0.697**	3.690	13.42	0.685	3.570	35.18	0.641	3.270	8.19	0.692	3.300	14.19	0.679	3.458
w/ zh2zh	2.76	0.692	3.660	13.90	**0.687**	3.550	36.65	0.644	3.260	8.92	0.694	3.320	15.06	0.679	3.448
w/ en2zh, zh2en	2.32	0.694	**3.700**	**11.78**	0.684	**3.580**	35.17	**0.645**	**3.290**	**7.00**	0.700	**3.330**	**14.07**	0.681	**3.475**
w/ all	2.35	0.695	3.680	13.76	0.686	3.560	**33.53**	0.642	3.240	7.38	**0.704**	3.310	14.26	**0.682**	3.448
On Chinese Evaluation Samples															
Model	**Regular (*zh*)**			**Articulatory (*zh*)**			**Code-switching (*zh2mixed*)**			**Cross-lingual (*en2zh*)**			**Avg**		
	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS	WER	SIM	UTMOS
ARS (Wang et al., 2025a)	4.37	0.752	2.730	24.07	0.711	2.430	59.71	0.756	2.900	24.30	0.563	3.090	28.61	0.696	2.788
w/ en2en	2.68	0.761	**2.760**	21.68	0.727	**2.530**	48.84	0.757	2.990	12.48	0.566	3.140	21.42	0.703	**2.855**
w/ zh2zh	**2.41**	0.760	2.740	**19.51**	**0.727**	2.490	47.99	0.755	**3.010**	12.73	0.565	3.110	20.16	0.702	2.838
w/ en2zh, zh2en	2.49	**0.762**	2.740	22.92	0.715	2.490	**41.00**	0.757	3.000	**11.76**	**0.573**	**3.160**	**19.54**	0.702	2.848
w/ all	2.62	0.759	2.720	21.06	0.725	2.440	41.50	**0.760**	2.980	11.95	0.572	3.090	19.78	**0.704**	2.808

Table 9: Effect of different languages within INTP for ARS. In these experiments, we use only the **Regular** part of INTP for training.

During the inference stage for zero-shot TTS, the input sequence is represented as:

$$[\boldsymbol{I}, \boldsymbol{T}_{\text{prompt}}, \boldsymbol{T}_{\text{target}}, <|\text{startofspeech}|>, \boldsymbol{S}_{\text{prompt}}]$$

to generate the target speech tokens $\boldsymbol{S}_{\text{target}}$. Here, $\boldsymbol{T}_{\text{prompt}}$, $\boldsymbol{T}_{\text{target}}$, $\boldsymbol{S}_{\text{prompt}}$ are placeholders for the prompt text, target text, and prompt speech tokens, respectively.

C.2 Training data

We pre-train Ints on Emilia (He et al., 2024), which consists of about 100K hours of multilingual data. Following this, we use INTP alignment to obtain Ints v1. Ints v1 is then used to generate new preference data, which are employed to train Ints v2 for iterative alignment. We select prompts from the repeated and code-switching samples of INTP, which can be considered a more challenging subset of prompts. For each prompt, we use the same INTP intra-pair pipeline in Appendix A.3.1 to construct preference pairs.

D Training Details

All of our experiments are conducted on 8 NVIDIA H100 80GB-GPUs. Unless stated otherwise, we use the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$ and train for one epoch. For each model, we provide more detailed information about the experiments:

- **ARS**: We use a learning rate of $5e-6$ with a warmup of $4,000$ steps and an inverse square root learning scheduler. For DPO, we use the hyperparameter $\beta = 0.1$.
- **F5-TTS**: We use a learning rate of $8e-6$ with a warmup of $4,000$ steps and an inverse square root learning scheduler. For DPO, we use the hyperparameter $\beta = 1,000$.
- **MaskGCT**: We use a learning rate of $5e-6$ with a warmup of $4,000$ steps and an inverse square root learning scheduler. For DPO, we use the hyperparameter $\beta = 10$.
- **CosyVoice 2**: We use a learning rate of $5e-6$ with a warmup of $4,000$ steps and an inverse square root learning scheduler. For DPO, we use the hyperparameter $\beta = 0.1$.
- **Ints**: We use a learning rate of $5e-6$ with a warmup of $4,000$ steps and an inverse square root learning scheduler. For DPO, we use the hyperparameter $\beta = 0.1$. We use flash attention (Dao et al., 2022) and bfloat16 for training.

E Additional Experimental Results

E.1 Effect of Data across Different Languages within INTP

We present the effect of different languages within INTP in Table 9. The results reveal three key findings: (1) Data from all languages can contribute to improvements across diverse domains for ARS. (2) Interestingly, using only English post-training data (w/ en2en) could also improve performance on Chinese evaluation samples, and vice versa, demonstrating that the proposed alignment algorithm enhances the model's foundation capability in intelligibility. (3) Furthermore, we again observe the effectiveness of preference alignment's customized feature: when aiming to improve performance on cross-lingual cases, directly constructing data from

Model	Japanese		Korean		German		French	
	WER	SIM	WER	SIM	WER	SIM	WER	SIM
Ints	26.34	0.714	31.67	0.708	28.25	0.674	54.53	0.545
w/ INTP	21.82	0.718	19.57	0.741	21.20	0.676	42.12	0.558

Table 10: Effect of INTP alignment for unseen languages.

the cross-lingual distribution yields the most significant gains.

E.2 Effect of INTP Alignment for Unseen Languages

We conducted the additional evaluations on four unseen languages not covered by INTP. Specifically, we tested the Ints models before and after INTP alignment using Japanese, Korean, German, and French speech data from GTSinger (Zhang et al., 2024d) (a dataset not used in either pre-training or post-training). We constructed evaluation sets consisting of 500 samples for each language. The results in Table 10 demonstrate that despite INTP containing only Chinese and English data, improvements in both WER and SIM metrics are observed across all four languages. We hypothesize that this generalization stems from our proposed intelligibility preference alignment method enhancing the model's fundamental capabilities in intelligibility such as the basic articulation and pronunciation.

F Evaluation Details

F.1 Evaluation Data

Our evaluation sets are based on SeedTTS test-en and SeedTTS test-zh datasets[7]. The SeedTTS test-en set includes 1,000 samples from the Common Voice dataset (Ardila et al., 2019), while the SeedTTS test-zh set comprises 2,000 samples from the DiDiSpeech dataset (Guo et al., 2021). We also provide the detailed distribution of our proposed sets in Table 11.

	Languages		#Total
Regular	en	zh	3,000
	1,000	2,000	
Articulatory	en	zh	800
	400	400	
Code-switching	en2mixed	zh2mixed	1,000
	500	500	
Cross-lingual	zh2en	en2zh	1,000
	500	500	

Table 11: Statistics of the proposed evaluation sets in four scenarios (**en**: English, **zh**: Chinese, **mixed**: mixture of English and Chinese, **zh2en**: Chinese reference speech with English target text. Similarly for **en2mixed**, **zh2mixed**, and **en2zh**).

F.2 Objective Evaluation Metrics

For objective metrics, we evaluate the intelligibility (WER), speaker similarity (SIM), and overall speech quality (UTMOS (Saeki et al., 2022)):

- **WER**: We employ `Whisper-large-v3`[8] (Radford et al., 2023) for English texts, and `Paraformer-zh`[9] (Gao et al., 2022, 2023) for Chinese and code-switching texts.
- **SIM**: We compute the cosine similarity between the WavLM TDNN[10] (Chen et al., 2022) speaker embeddings of generated samples and the prompt samples.
- **UTMOS**: We use the pretrained UTMOS strong learner following the official implementation[11].

F.3 Subjective Evaluation

We consider four different settings: regular, articulatory, code-switching, and cross-lingual. Each setting is evaluated in two languages, resulting in 10 samples per language. This setup yields a total of 80 pairs. These 80 pairs are evaluated across 5 different systems (ARS, F5-TTS, MaskGCT, CosyVoice 2, and Ints), leading to a total of 400 pairs. We engage 20 participants in the evaluation process, ensuring that each sample is assessed at least three times.

We conduct subjective evaluations from three perspectives: intelligibility (reading accuracy), naturalness (N-CMOS), and speaker similarity (A/B

[7] https://github.com/BytedanceSpeech/seed-tts-eval
[8] https://huggingface.co/openai/whisper-large-v3
[9] https://huggingface.co/funasr/paraformer-zh
[10] https://github.com/microsoft/UniSpeech/tree/main/downstreams/speaker_verification
[11] https://github.com/sarulab-speech/UTMOS22

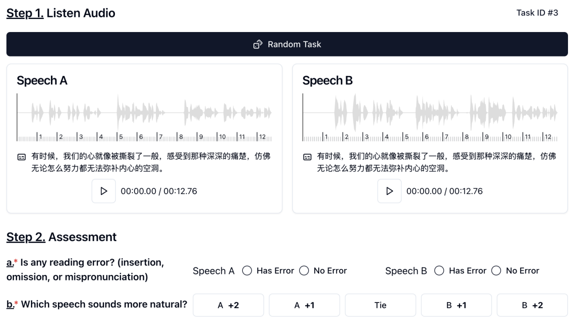


Figure 3: User interface for intelligibility and naturalness evaluation.

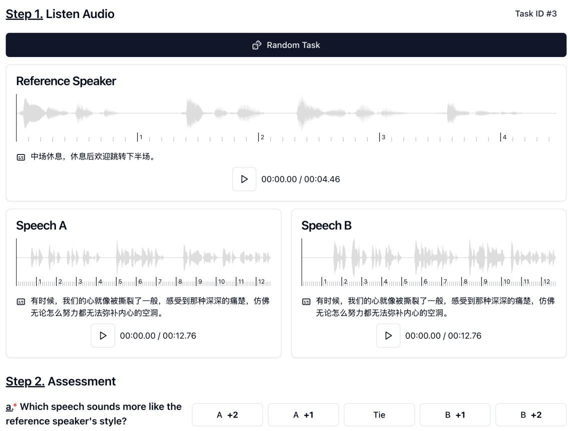


Figure 4: User interface for speaker similarity evaluation.

Testing). We have developed an automated subjective evaluation interface, as shown in Figure 3 and Figure 4. For each item to be evaluated, users will see three components: the System Interface, the Questionnaire, and the Evaluation Criteria.

Intelligibility (Reading Accuracy):

- **System Interface:** Users listen to the speech audio and compare it to the provided target text to assess whether the speech matches the text.
- **Questionnaire:** Users are asked, "Is any reading error? (insertion, omission, or mispronunciation)"
- **Evaluation Criteria:** The evaluation is binary: "No Error" (the speech matches the text) or "Has Error" (the speech does not match the text).

Naturalness (N-CMOS):

- **System Interface:** Users listen to two speech samples, A and B, to compare their naturalness.
- **Questionnaire:** Users are asked, "Which speech sounds more natural?"
- **Evaluation Criteria:** Options include A +2 (Sample A is much more natural), A +1 (Sample A is slightly more natural), Tie (Both are equally natural), B +1 (Sample B is slightly more natural), and B +2 (Sample B is much more natural).

Speaker Similarity (A/B Testing):

- **System Interface:** Users listen to two speech samples, A and B, to evaluate their similarity to the speech of the reference speaker.
- **Questionnaire:** Users are asked, "Which speech sounds more like the reference speaker's style?"
- **Evaluation Criteria:** Options include A +2 (Sample A is much more similar), A +1 (Sample A is slightly more similar), Tie (Both are equally similar), B +1 (Sample B is slightly more similar), and B +2 (Sample B is much more similar).