SpeechAlign: Aligning Speech Generation to Human Preferences

Dong Zhang[*], **Zhaowei Li**[*], **Shimin Li**, **Xin Zhang**, **Pengyu Wang**,
Yaqian Zhou[†], **Xipeng Qiu**[†]
Fudan University
`dongzhang22@m.fudan.edu.cn`, `lizhaowei126@gmail.com`
`{zhouyaqian,xpqiu}@fudan.edu.cn`

`https://0nutation.github.io/SpeechAlign.github.io/`

Abstract

Speech language models have significantly advanced in generating realistic speech, with neural codec language models standing out. However, the integration of preference optimization to align speech outputs to human preferences is often neglected. This paper addresses this gap by first analyzing the distribution gap in codec language models, highlighting how it leads to discrepancies between the training and inference phases, which negatively affects performance. Then we explore leveraging preference optimization to bridge the distribution gap. We introduce SpeechAlign, an iterative self-improvement strategy that aligns speech language models to human preferences. SpeechAlign involves constructing a preference codec dataset contrasting golden codec tokens against synthetic tokens, followed by preference optimization to improve the codec language model. This cycle of improvement is carried out iteratively to steadily convert weak models to strong ones. Through both subjective and objective evaluations, we show that SpeechAlign can bridge the distribution gap and facilitating continuous self-improvement of the speech language model. Moreover, SpeechAlign exhibits robust generalization capabilities and works for smaller models. Demos are available at `https://0nutation.github.io/SpeechAlign.github.io/`.

1 Introduction

Large language models (LLMs) have showcased their potent abilities through techniques such as pretraining, supervised fine-tuning (SFT), and Reinforcement Learning from Human Feedback (RLHF) (OpenAI, 2023; Touvron et al., 2023). The field of speech language modeling has seen significant progress (Wang et al., 2023; Borsos et al., 2022; Zhang et al., 2023a), particularly with the adoption of discrete speech representations (Hsu et al., 2021; Zhang et al., 2023b) like audio codecs (Défossez et al., 2022; Zeghidour et al., 2021; Zhang et al., 2023d). However, current speech language models primarily focus on the SFT stage associated with empowering the LLM's instruction-following capabilities, neglecting the integration of human feedback to align speech outputs to human preferences regarding quality, naturalness, and expressiveness. Fortunately, *preference optimization* has emerged as a powerful solution for aligning LLM output distribution with human expectation (Stiennon et al., 2022; Bai et al., 2022; Ouyang et al., 2022). The most successful approach, RLHF, achieves this by integrating rewarding modeling and a reinforcement learning phase.

[*]Equal contribution.
[†]Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Additionally, some computationally efficient alternatives have proven to be effective in aligning LLM behavior without the need for explicit reward modeling (Rafailov et al., 2023; Zhang et al., 2023c; Wang et al., 2024).

The key to the success of speech language models (Wang et al., 2023; Borsos et al., 2022; Zhang et al., 2023a) that build on LLMs is utilizing audio codecs that discretize the speech representations. Neural Codec Language Models, leveraging audio codecs, have demonstrated their effectiveness in speech generation tasks (Yang et al., 2023; Wang et al., 2023). It primarily utilizes a hierarchical approach that consists of a pipeline of autoregressive (AR) and non-autoregressive (NAR) models, as illustrated in Figure 2 (a). AR model generates semantic tokens (Borsos et al., 2022) or the first layer of codec tokens (Wang et al., 2023), referred to as AR tokens. These AR tokens serve as input for NAR model to generate acoustic tokens (Borsos et al., 2022) or subsequent layers of codec tokens (Wang et al., 2023), termed as NAR tokens. However, this pipeline system introduces a discrepancy between the training and inference phases for the codec language model. In training, NAR model is fed with **golden AR tokens** derived from real speech, but the model receives **synthetic AR tokens** generated by the AR model during inference. As demonstrated in Section 2.3, there is a distribution gap between these two types of AR tokens, which adversely impacts the performance of the NAR model.

Can we calibrate the output of codec language models to the authentic codec distribution by preference optimization? Collecting a large, high-quality preference dataset for codec language models is challenging. First, codec tokens are often represented in numerical form, which is not directly understandable by humans, making it impossible to collect human preferences for these tokens directly. Furthermore, collecting human preferences on speech to gather feedback on codec tokens poses multiple challenges, including inconsistency across various human annotators and the difficulty of scaling up the dataset size.

We propose SpeechAlign, an iterative self-improving strategy that aligns speech language models to human preferences. To avoid the need for additional human-annotated data, we construct the pairwise preference codec dataset by considering golden AR tokens as preferred data and synthetic AR tokens as dis-preferred data. Human verification is conducted to ensure its consistency with human preferences. After obtaining the preference dataset, we explore different preference optimization strategies to improve codec language models. Following a complete cycle, we iteratively perform preference dataset collection and preference-aware optimization to convert weak codec language models to stronger ones continually. Experimental results show that SpeechAlign can continually improve the speech generation performance of speech language models. Our contributions are summarized below: 1) We propose SpeechAlign, the first to align speech language models by preference optimization. 2) We propose an iterative self-improving strategy to convert weak codec language models to stronger ones without additional human-annotated data. 3) We analyze the issue of distribution gaps in codec language models and explore various strategies to bridge the gap.

2 Preliminary Analysis on Distribution Gap

In this section, we perform preliminary experiments to analysis the distribution gap between golden and synthetic codec tokens and demonstrate that this gap can degrade the performance of the codec language models.

2.1 Background

We build a codec language model, referred to as *SpeechAlign-sft*, serving as the baseline system to analysis the distribution gap. Similar to (Zhang et al., 2024; Budzianowski et al., 2024), we rely on SpeechTokenizer (Zhang et al., 2023d) to extract speech codec tokens. SpeechTokenizer is a Residual Vector Quantization (RVQ)-based speech tokenization method and hierarchically disentangles different aspects of speech information across different RVQ layers. The output of SpeechTokenizer comprises $Q = 8$ hierarchical RVQ tokens $(q_1, \ldots, q_Q)$. *SpeechAlign-sft* consists of a SpeechGPT (Zhang et al., 2023a)-based autoregressive (AR) model and a SoundStorm (Borsos et al., 2023)-based non-autoregressive (NAR) model. The AR model learns the mapping from input golden text to the first layer of codec tokens q_1. We continue finetuning the pretrained SpeechGPT model in (Zhan et al., 2024) on LibriSpeech dataset to get the AR model. Details about training process as described in Section 4.1. The NAR model adopts the training and inference procedure of SoundStorm (Borsos et al., 2023) and learns to generate subsequent layers of SpeechTokenizer

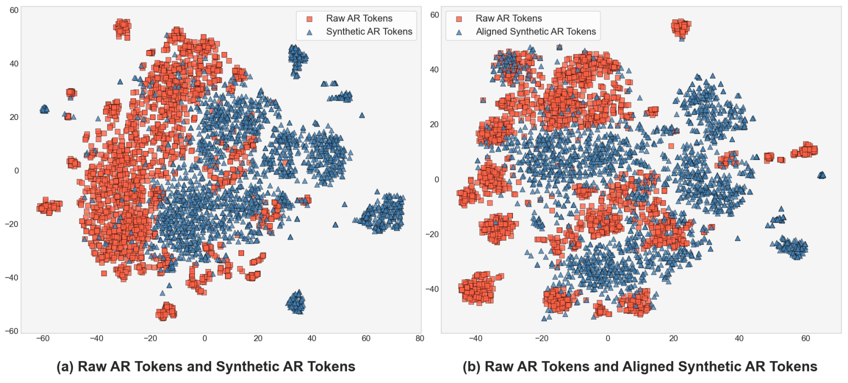


Figure 1: T-SNE visualization of representations of different AR tokens. **Left**: Golden AR tokens and synthetic AR tokens. **Right**: Golden AR tokens and aligned synthetic AR tokens.

tokens conditioning on the first layer tokens and prompt speech. We use the pretrained SoundStorm model in (Zhan et al., 2024). At inference time, the AR model converts input text to **AR tokens** and the NAR model uses these tokens along with prompt speech as conditions to generate **NAR tokens**. These tokens are then concatenated and converted into speech by the SpeechTokenizer decoder.

2.2 Visualization of Distribution Gap

To analysis the distribution gap, we randomly select 1000 speech-text pairs from LibriSpeech dataset and construct a test corpus composed of triplets $D_{vis} = \{(\mathbf{t}, \mathbf{y_g}, \mathbf{y_s})\}$ following the procedure in Section 3.1. Here $\mathbf{t}$ is the input text, $\mathbf{y_g}$ is the golden AR tokens and $\mathbf{y_s}$ is the synthetic AR tokens generated by SpeechAlign-sft. The input text $\mathbf{t}$ is concatenated with the golden AR tokens $\mathbf{y_g}$ and fed into the SpeechAlign-sft model. This process yields the hidden states of the last layer for each AR token in the sequence. By applying mean pooling across the temporal dimension, these hidden states are aggregated to produce a single vector representation Rep_g for the golden AR tokens. Similarly, we acquire Rep_s for the synthetic AR tokens using the same procedure. The vectors are visualized in a 2D space using t-SNE, as shown in Figure 1 (a). We can observe that the representations of golden AR tokens and synthetic AR tokens are so dissimilar that they naturally form two distinct clusters, indicating that significant distributional gap exists between them.

2.3 Distribution Gap Degrades Performance

The NAR model is trained using golden AR tokens as input, but during inference, the input switches to synthetic AR tokens. This results in a discrepancy between the training and inference processes due to the existing distribution gap, potentially affecting performance. To delve into this issue, we conduct a speech reconstruction experiment with NAR model. We construct a dataset composed of triplet data $D_{test} = \{(\mathbf{z}, \mathbf{y_g}, \mathbf{y_s})\}$, with $\mathbf{y_g}$ and $\mathbf{y_s}$ described in Section 2.2 and $\mathbf{z}$ represents 3-second prompt speech from the same speaker but distinct from the speech used for $\mathbf{y_g}$ and $\mathbf{y_s}$. The NAR model performs speech reconstruction by taking prompt speech combined with either golden AR tokens or synthetic AR tokens as input, to generate speech for each type of tokens respectively. The quality of the generated speech is evaluated based on the word error rate (WER) and speaker similarity (SIM) metrics, compared against the ground truth. As shown in Tabel 1, speech generated from golden AR tokens exhibits superior WER and Speaker Similarity scores compared to that generated from synthetic AR tokens. This finding proves that the distribution gap adversely affects the NAR model's performance.

Input	WER(↓)	SIM(↑)
Groundtruth	3.4	-
Golden AR tokens	5.9	0.93
Synthetic AR tokens	7.2	0.87

Table 1: Results of NAR model's speech reconstruction performance with different AR tokens as input.

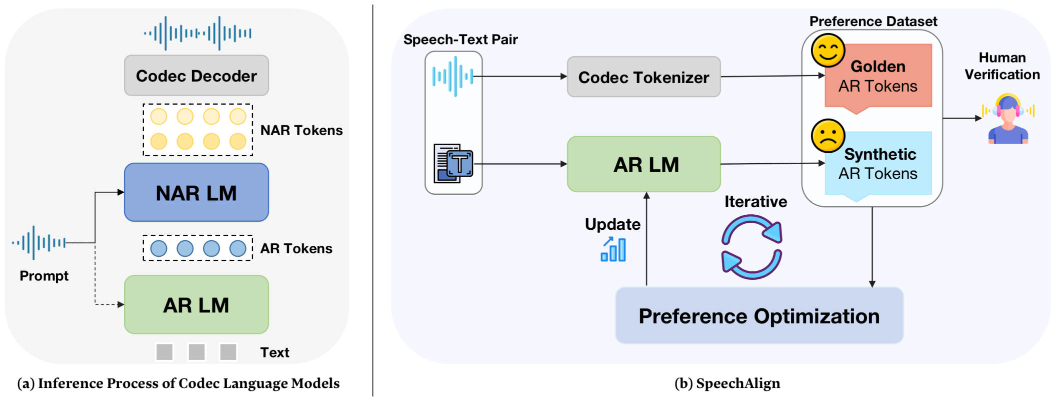


Figure 2: AR LM refers to autoregressive models and NAR LM refers to non-autoregressive models. **Left**: Illustration of inference process of codec language models. **Right**: Illustration of SpeechAlign method.

3 SpeechAlign

We take SpeechAlign-sft detailed in Section 2.1 as the baseline system, referred to as p_{θ_0}. Within this framework, the AR model is represented as $p^{ar}_{\theta_0}$, and the NAR model as $p^{nar}_{\theta_0}$. As shown in Figure 2 (b), the first step of SpeechAlign is to construct perference dataset that contrasts golden codec tokens with synthetic codec tokens. Utilizing this dataset, we implement various preference optimization strategies to align the baseline model. This process is iteratively executed, enabling the continuous self-improvement of codec language models.

3.1 Preference Data Collection

Golden Win	Tie	Golden Lose
71%	21%	8%

Table 2: Comparison between reconstructed speech from golden AR tokens versus synthetic AR tokens.

A standard method for collecting preference dataset involves prompting the model to produce two distinct responses to a query, after which annotators are asked to select the one they prefer. However, collecting human preferences for codec data is impractical and unscalable. Instead, we construct the preference codec dataset by contrasting the golden codec tokens against synthetic codec tokens. Concretely, we randomly sample N speech-text pairs $P = \{(\mathbf{s}, \mathbf{x})\}_{i=1}^{N}$ from LibriSpeech dataset, where $\mathbf{s} = (s_1, ..., s_{|s|})$ is the speech and $\mathbf{x} = (x_1, ..., x_{|x|})$ is the corresponding transcript and N is 50000. For each speech $\mathbf{s}$, we adopt pretrained SpeechTokenizer to extract discrete representations and denote the tokens of first RVQ layer as golden AR tokens $\mathbf{y_g}$. For the corresponding transcript $\mathbf{x}$, the AR model $p^{ar}_{\theta_0}$ takes it as input to generate synthetic AR tokens $\mathbf{y_s}$. Following these steps, we can get the preference codec dataset $D_{pf} = \{(\mathbf{x}, \mathbf{y_g}, \mathbf{y_s})\}_{i=1}^{N}$.

Human Verification To validate the quality of constructed preference codec dataset, we perform human verification by randomly sampling 100 entries from D_{pf} and employing the same procedure outlined in section 2.3 to convert y_g and y_s back into speech. This allows humans to compare them side by side and choose the better speech in terms of both speech quality and voice similarity. From results in Table 2, we can conclude that humans prefer speech reconstructed from golden AR tokens over that from synthetic AR tokens, indicating that the constructed preference codec dataset effectively aligns with human preferences.

3.2 Preference Optimization

In this section, we introduce how we conduct preference optimization to align codec language models using preference codec dataset, including Chain-of-Hindsight (Liu et al., 2023), Direct Preference Optimization (Rafailov et al., 2023), RLHF-PPO (Ouyang et al., 2022) and Best-of-N Sampling.

Chain-of-Hindsight (CoH) By converting various forms of feedback into sentences and integrating these with the respective responses, CoH enables models to learn from both positive and negative

feedback, allowing the identification and correction of negative attributes or errors. At inference time, the model is guided to generate the desired outputs according to the feedback type in prompt. In our case, we first convert feedback signals into a descriptive template and construct training data by combining responses with corresponding feedback template as follows:
$\mathbf{T_g}$ = "*[Human]: Read this text and give me a high-quality speech response:* $\{\mathbf{x}\}$ *<eoh> [SpeechGPT]:* $\{\mathbf{y_g}\}$ *<eoa>*."
$\mathbf{T_s}$ = "*[Human]: Read this text and give me a low-quality speech response:* $\{\mathbf{x}\}$ *<eoh> [SpeechGPT]:* $\{\mathbf{y_s}\}$ *<eoa>*."
The AR model is optimized via the negative log-likelihood loss on preference corpus D_{pf} as follows:

$$L_{COH} = -\mathbb{E}_{(x,y_g,y_s)\sim D_{pf}}[\log p^{ar}_{\theta_0}(y_g|x,T_g) + \log p^{ar}_{\theta_0}(y_s|x,T_s)]$$

During inference phrase, we prompt the model with positive feedback in the form of 'high-quality' to guide the model in generating favorable outputs.

Direct Preference Optimization (DPO) Without using explicit reward modeling or reinforcement learning, DPO can fine-tune the model to align with human preferences. DPO considers the likelihood of preferred response over dispreffered response and optimizes the LLM model towards that objective. The prompt template for DPO training is as follows:
$\mathbf{T}$ = "*[Human]: Read this text and give me a speech response:* $\{\mathbf{x}\}$ *<eoh> [SpeechGPT]:* $\{\mathbf{y}\}$ *<eoa>*. "
In our case, the DPO loss can be formated as follows:

$$L_{DPO} = -\mathbb{E}_{(x,y_g,y_s)\sim D_{pf}}[\log\sigma(\log\frac{p^{ar}_{\theta}(y_g|x,T)}{p^{ar}_{ref}(y_g|x,T)} - \log\frac{p^{ar}_{\theta}(y_s|x,T)}{p^{ar}_{ref}(y_s|x,T)})]$$

where p^{ar}_{ref} is the reference model and initialize with p^{ar}_{θ}.

RLHF-PPO RLHF methods involve training a reward model on a dataset reflecting human preferences. RL algorithms are then applied to adjust a language model's policy to favor responses that are highly rewarded, while ensuring minimal deviation from the original model's behavior. With preference dataset D_{pf}, we can parameterize a reward model $r_\phi(x,y)$ and estimate the parameters via maximum likelihood. By treating the task as a binary classification, we utilize the negative log-likelihood loss:

$$L_{rm} = \mathbb{E}_{(x,y_g,y_s)\sim D_{pf}}[\log\sigma(r_\phi(x,y_g) - r_\phi(x,y_s)]$$

where σ is the logistic function. The reward model $r_\phi(x,y)$ is initialized from AR model p^{ar}_{θ} with a linear layer atop the last Transformer layer to yield a single scalar prediction as the reward value. During the RL stage, we optimize the AR model against the reward model using PPO algorithm. Specially, we refine the AR model $p^{ar}_{\theta_0}$ as the following optimization problem:

$$\max_{p^{ar}_{\theta_0}} \mathbb{E}_{x\sim D_{pf},y\sim p^{ar}_{\theta_0}(y|x)}[r_\phi(x,y)] - \beta\,\mathbb{D}_{kl}[\,p^{ar}_{\theta_0}(y|x)||p^{ar}_{ref}(y|x)\,]$$

where β represents a coefficient regulating the extent of the KL penalty and p^{ar}_{ref} is the reference model and initialize with p^{ar}_{θ}.

Best-of-N Sampling (BoN) With the reward model trained on the preference data, we implement a Best-of-N approach to enhance the quality of output codec tokens. Concretely, we sample N responses using the AR model. These responses are then evaluated by the reward model, and the one receiving the highest reward score is chosen as the final response to serve as input for NAR model.

3.3 Iterative Self-Improvement

Following the aforementioned steps results in an updated AR model, denoted as $p^{ar}_{\theta_1}$. Using this updated model, we can create a new preference codec dataset, D_{pf}. This dataset then serves as the basis for further improvement of the AR model through preference optimization. The iterative self-improvement process of the AR model, as detailed in algorithm 1, enables continuous calibration of the output distribution towards the authentic codec token distribution.

Algorithm 1 SpeechAlign

```
Input: $\{(s_i, x_i)\}_{i=0}^N$: Speech-Text Dataset, $m_\phi$: pretrained SpeechTokenizer model with parameter $\phi$, $p_{\theta_0}^{ar}$: AR model with parameter $\theta_0$, $T$: Number of iterations.
for $t = 0, \dots, T-1$ do
    for $i = 1, \dots, N$ do
        Generate golden AR token $y_{r_i} \sim m_\phi(\cdot|s_i)$
        Generate synthetic AR token $y_{s_i} \sim p_{\theta_t}^{ar}(\cdot|x_i)$
    end for
    Update $\theta_{t+1}$ by performing preference-aware optimization on $\theta_t$ with $D_{pf_i} = \{(x_i, y_{r_i}, y_{s_i})\}_{i=1}^N$
end for
Output: $\theta_T$.
```

4 Experiments

4.1 Setups

Data For the continue finetuning stage in Section 2.1, we use the LibriSpeech dataset. To construct the preference codec dataset, we randomly sample 50k speech-text pairs from LibriSpeech training set. During the iterative process of SpeechAlign, we utilize the synthetic data generated in the most recent iteration and combine it with the newly produced synthetic data. As a result, the size of the synthetic dataset increases across iterations: starting at 50k in iteration 0, and expanding to 100k in iterations 1, 2, and 3.

Model For the AR model, we further finetune the pretrained SpeechGPT model in (Zhan et al., 2024) on LibriSpeech dataset. For the NAR model, we use the pretrained SoundStorm model in (Zhan et al., 2024).

Training For the continue finetuning stage in Section 2.1, the batch size is set to 256, with a learning rate of 1e-5 and train for 3500 steps on 8 A100 80G GPUs. For CoH finetuning, the batch size is set to 32, with a learning rate of 1e-5 and train for 12000 steps on 8 A100 80G GPUs. For DPO finetuning, the batch size is set to 128, with a learning rate of 5e-7 and train for 2000 steps on 8 A100 80G GPUs. For reward model training, the batch size is set to 32, with a learning rate of 1e-5 and train for 1000 steps on 8 A100 80G GPUs. For PPO training, the batch size is set to 16, with a learning rate of 1e-5 and train for 1000 steps on 8 A100 80G GPUs.

4.2 Evaluation and Metrics

We conduct zero-shot TTS evaluation on LibriSpeech test-clean set and VCTK dataset. For each speaker, we randomly selected a 3s utterance as the prompts while the textual content of a different utterance is used as the input text. To reduce the randomness in the evaluation process, we evaluate each model ten times and then calculate the average to obtain the final result. The metrics we adopt are as follows:

Word Error Rate (WER) is utilized to assess the content accuracy of synthesized speech by calculating the distance between the synthesized speech's transcription and the input text. We use Whisper medium-en model Radford et al. (2022) to transcribe the synthesized speech.

Speaker Similarity (SIM) evaluates the consistency of timbre between the synthesized and the prompt speech. This is measured by the similarity between the speaker embedding of generated speech and that of the speech prompt. The similarity calculation involves the following steps: 1) employing a speaker embedding extractor [3] to derive the speaker embeddings for both the generated and prompt speech, and 2) computing the cosine similarity between these normalized embeddings.

Human Evaluation We conduct comparative testing of various models' outputs against the baseline system's speech. The evaluators are provided with prompt speech, the baseline system's speech, and our model's speech. Human evaluators are tasked with determining which utterance sounded more natural and closer to the prompt speech. Evaluators have the option to choose either of the two utterances or indicate that they perceive them as equally natural. We also evaluate the QMOS

[3] `https://huggingface.co/microsoft/wavlm-base-plus-sv`.

Model	LibriSpeech				VCTK			
	WER (↓)	SIM (↑)	QMOS (↑)	SMOS (↑)	WER (↓)	SIM (↑)	QMOS (↑)	SMOS (↑)
Groundtruth	4.0	-	4.40	4.06	2.4	-	4.33	4.60
Baselines								
SpeechAlign-sft	7.2	0.87	3.20	3.20	8.8	0.79	3.27	3.13
Continue SFT	8.0	0.88	3.07	3.13	9.8	0.80	3.20	3.20
SpeechAlign-CoH	7.3	0.89	3.33	3.47	10.2	0.81	3.53	3.73
SpeechAlign-BoN	8.0	0.88	3.40	3.70	**7.5**	0.79	3.50	3.40
SpeechAlign-RLHF-PPO	7.1	0.89	3.60	3.87	8.5	0.80	**3.53**	**3.80**
SpeechAlign-DPO-Iter1	6.7	0.88	3.20	3.33	8.5	0.82	3.33	3.07
SpeechAlign-DPO-Iter2	6.2	0.89	3.67	3.40	8.0	0.83	3.33	3.33
SpeechAlign-DPO-Iter3	**6.0**	**0.90**	**3.73**	**3.93**	7.9	**0.83**	3.47	3.60

Table 3: Evaluation Results of zero-shot text-to-speech on LibriSpeech and VCTK. Each result is calculated as the average of ten separate evaluations.

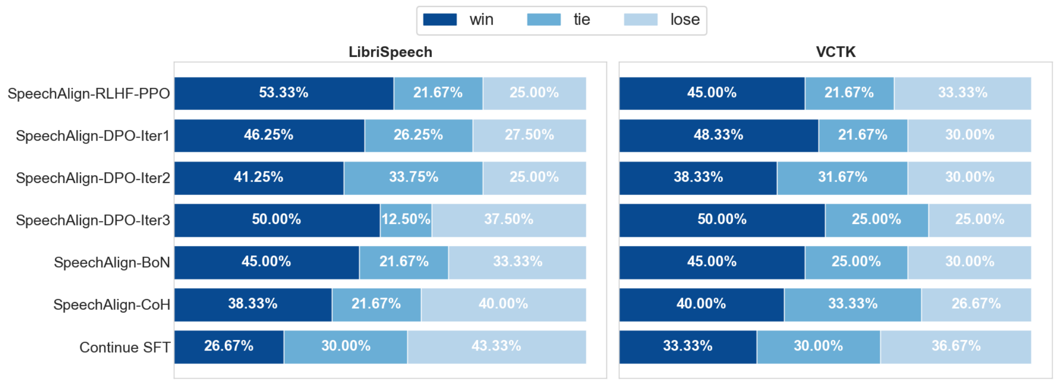


Figure 3: Qualitative side-by-side comparsion results of preference optimized models versus the baseline SFT model on zero-shot text-to-speech performance.

and SMOS score. MOS reflects the quality and naturalness of speech. SMOS reflects the timbre similarity between speech prompts and generated speech. They both span from 1 to 5, with higher values signifying better results. Each evaluation receives 6 ratings from 6 different human evaluators.

4.3 Main Results

Preference Optimization Boosts Speech Generation For objective evaluation, Table 3 shows that the WERs of SpeechAlign-RLHF-PPO and SpeechAlign-DPO series models are lower than that of SpeechAlign-sft. This suggests that preference optimization can enhance the accuracy of content modeling. Furthermore, these models also achieved superior performance in Speaker Similarity, indicating that preference optimization can also improve the effectiveness of timbre modeling. For subjective evaluation, our preference-optimized models all achieve higher QMOS and SMOS scores than SpeechAlign-sft. Figure 3 reveals that our preference-optimized models, SpeechAlign-BoN, SpeechAlign-RLHF-PPO and SpeechAlign-DPO-Iter1, significantly outperform the baseline model in win rates. These findings underscore the effectiveness of SpeechAlign in significantly improving the capabilities of codec language models across content, timbre, and audio quality dimensions.

Speech Language Model Can Self-Improve Iteratively The quantitative results in Table 3 show that from Iteration 1, DPO contributes to enhancements in speech generation. Iteration 2 further amplifies these improvements, with a notably significant impact on the WER. And the trend of gradual enhancement is maintained in subsequent iterations. By Iteration 3, there is a reduction in WER by 0.8 compared to the Baseline, and Speaker Similarity has increased to 0.9. The same trend can be observed in QMOS and SMOS metrics. Figure 3 shows that SpeechAlign-DPO-Iter3 achieves a higher win rate compared to SpeechAlign-DPO-Iter1, indicating superior performance in qualitative evaluation. This confirms that iterative DPO can consistently enhance the quality of the speech generated by the model. It demonstrates that SpeechAlign is an effective method for the speech language model to undergo continuous and efficient self-improvement.

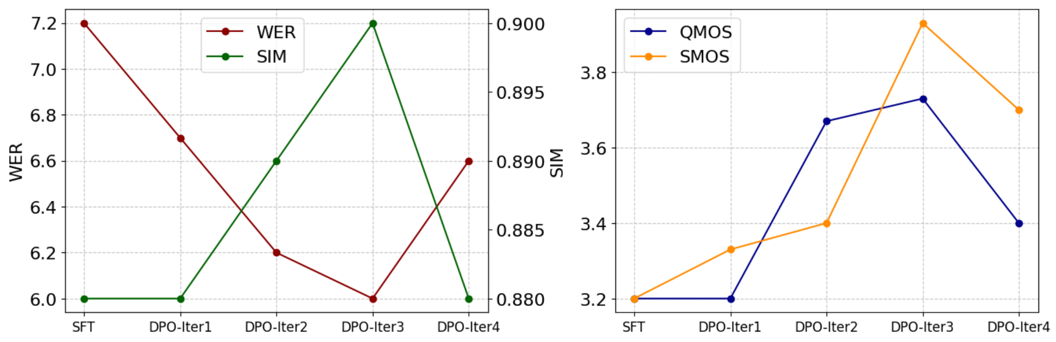


Figure 4: Zero-shot TTS performance of DPO-optimized models at different iterations on LibriSpeech test-clean.

Generalization to Unseen Speakers We also evaluate whether SpeechAlign would bring better speech generation when encountering unseen speaker in the preference data. We evaluate different models' performances on VCTK dataset, which have no speaker overlap with the training dataset. As shown in Table 3, SpeechAlign-RLHF-PPO, SpeechAlign-BoN, and SpeechAlign-DPO can still improves the generated speech across all metrics. We also observe similar improvements in subjective evaluation in Figure 3. And iterative optimization can bring continuous improvement, suggesting that SpeechAlign can be generalized to unseen speakers. We can observe that SpeechAlign-RLHF-PPO outperforms the SpeechAlign-DPO series in both QMOS and SMOS metrics, suggesting that models optimized by RLHF generalize to other domains more effectively than those optimized with DPO.

5 Analysis

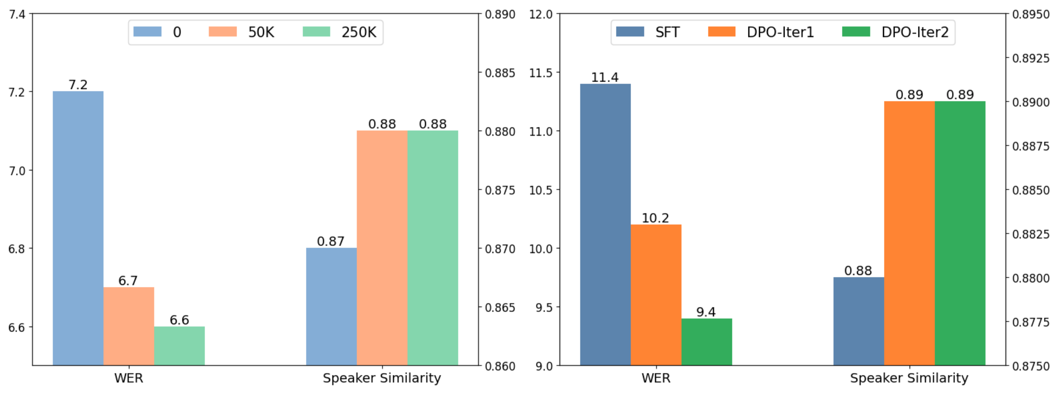
5.1 Iterative Self-Improvement has a Upperbound

To investigate whether there is a limiting point for iterative self-improvement, we conduct the fourth iteration DPO optimization based on SpeechAlign-DPO-Iter3. As shown in Figure 4, we can observe that iterative DPO optimization can bring continuous improvement until the third iteration. Additionally, the improvements observed in the second and third iterations are significantly greater than those in the first iteration. This might be because the negative samples (synthetic AR tokens) in the first iteration are of lower quality and have a larger gap compared to the positive samples, resulting in less effective preference optimization. As the model's performance improve, the gap between negative and positive samples decreases, making preference optimization more effective. However, in the fourth iteration, the model's performance decreases, possibly because the negative samples are of too high quality and too similar to the positive samples, increasing the difficulty of implicit reward modeling and preference optimization and even reducing the model's performance.

5.2 Ablation Studies

Preference Data Size We examine the effect of different training data sizes on the performance of SpeechAlign. We set the preference data size to be 0, 50k and 250k and generate the data accordingly. We make sure that the larger dataset includes the smaller ones. After one epoch of DPO fine-tuning for each of these training sizes, we assess their performance in zero-shot TTS. From Figure 5 (a), we can observe notable improvement across all metrics with increasing training sizes from 0 to 50k, indicating that an increase in preference data can enhance the effectiveness of SpeechAlign. However, employing 250k preference data through DPO does not result in larger performance gains, suggesting that there is a threshold beyond which additional data does not translate into better learning outcomes. This effect suggests that while increasing the size of preference data can initially lead to more effective feedback learning, there comes a point where the quality of data or the model's ability to utilize this data effectively becomes more critical than sheer volume.

Comparison With Continue SFT The chosen samples in the preference dataset originate from ground truth AR tokens, therefore, during the preference optimization process, the model undergoes retraining on the data. To investigate whether the improvements from preference optimization or simply from continued training on ground truth data, we conduct experiments on continuing



(a) Performance across Different Preference Data Sizes

(b) Performance on Small Models

Figure 5: **Left**: Performance of SpeechAlign across different preference data sizes. **Right**: Performance of SpeechAlign on small models. Results are evaluated from on LibriSpeech test-clean.

Supervised Fine-Tuning (SFT) from SpeechAlign-sft for comparison. We utilize the same ground truth AR tokens as those in the preference dataset, with training settings consistent with the DPO training described in Section 4.1. The findings presented in Table 3 and Figure 3 reveal that continuing SFT does not enhance the quality of generated speech; rather, it may actually degrade performance. This indicates that the capability improvements gained from SpeechAlign are attributable to preference-aware learning.

5.3 SpeechAlign Works With Small Models

Current codec language models primarily rely on smaller AR models with fewer than 1 billion parameters (Wang et al., 2023; Borsos et al., 2022). So we investigate whether SpeechAlign can bring improvements for smaller AR models. We utilize an AR model with 130 million parameters that features a 12-layer transformer decoder-only architecture with embedding dimension 768 and 16 heads. We train the AR model on Multilingual LibriSpeech dataset for 500000 steps with batch size 1152 and learning rate 1e-5. We adopt the NAR model in Section 2.1. For DPO training, the preference data size is 50k and we following the training setting in Section 4.1. We evaluate the zero-shot TTS performance on LibriSpeech test-clean dataset. Figure 5 (b) shows that in the first iteration, DPO decreases the WER from 11.4 to 9.3 and boosts Speaker Similarity from 0.87 to 0.88. This indicates that SpeechAlign can notably improve speech generation in smaller models. As the process advances, it's evident that the effectiveness of DPO isn't just limited to initial gains. With subsequent iterations, we observe a consistent upward trend in WER though no more improvement in speaker similarity.

5.4 Preference Optimization Bridges the Distribution Gap

As depicted in Figure 1 (b), after alignment, the representations of golden AR tokens and synthetic AR tokens merge into a single cluster without significant distribution differences. This demonstrates that the distribution gap can be bridged by preference optimization. Along with the results in Table 3 and Figure 3, we observe that as the distribution gap diminishes, the model's capability in speech generation improves. This proves the effectiveness of preference optimization in calibrating model's output distribution. By reducing the inconsistencies between the training and inference phases, the model can more accurately capture the features of the target distribution, resulting in more natural and accurate speech generation.

6 Related Work

Neural Codec Language Models AudioLM (Borsos et al., 2022) is the pioneering model in introducing codec codes for language modeling, adopting a hierarchical strategy that combines semantic and acoustic modeling. VALL-E (Wang et al., 2023), another innovative neural codec language model, is trained to produce discrete codes based on EnCodec (Défossez et al., 2022), enabling the generation of high-quality, personalized speech from just a 3-second sample of an unseen speaker's speech. However, all existing codec language models have been trained through supervised methods. SpeechAlign is the first work enable codec language models to learn from human feedback.

Learning From Human Feedback Learning from human feedback recently became a critical step in the LLM training such as ChatGPT (OpenAI, 2023) and LLaMA (Touvron et al., 2023). Existing human preference alignment methods for LLM include RLHF (Ouyang et al., 2022; Bai et al., 2022), contrastive learning (Rafailov et al., 2023) and Chain-of-Hindsight (Liu et al., 2023). MusicRL (Cideron et al., 2024) aligns music generation to human preferences by reinforcement learning, Baton (Liao et al., 2024) improves audio generation by learning from human feedback and (Lee et al., 2023) aligns text-to-image models using human feedback. (Liu et al., 2021) proposes to improve emotional text-to-speech via reinforcement learning. However, our work explores using human feedback to align codec language models for speech generation with human preference.

Self-Improvement SPIN Chen et al. (2024) enables the LLM to self-improve without additional human data or feedback from stronger LLMs by generating its own training data from its previous iterations and refining its policy by discerning these self-generated responses from those obtained from human-annotated data. SPIN-Diffusion Yuan et al. (2024) makes the diffusion model engage in competition with its earlier versions, facilitating an iterative self-improvement process. (Xu et al., 2024) adopts RLHF to improve LLM translation quality by optimizing reward models by distinguishing between human and machine translations. Similarly, our work converts weak speech language models stronger learners through self-improvement iteratively.

7 Conclusion

This paper first analyzes the distribution gap existing in current neural codec language models and propose to solve it by preference optimization. To avoid the need for additional human-annotated preference data, construct a preference codec dataset contrasting golden codec tokens against synthetic tokens. Then we conduct preference optimization to align codec language models to human preference. Subjective and objective evaluation results prove the effectiveness of SpeechAlign to continuously converting weak codec language models to stronger ones.

8 Limitations and Related Work

Fine-grained Reward Signals from Real-World Human Preferences Current preference datasets capture overall preferences, while speech preferences can be multi-faceted, including aspects like sound quality, rhythm, and timbre. Considering human preferences from these various dimensions can enhance speech generation capabilities in a more detailed manner. Additionally, gathering high-quality, real human preference data could be more effective than the current methods of preference dataset collection, as it allows for a nuanced understanding of user preferences that can lead to more targeted and efficient improvements in speech generation technologies.

Preference Optimization of the NAR Models In current practices, preference optimization is employed to enhance the capabilities of AR models. Nonetheless, the codec decoder also grapples with inconsistency problems during training and inference, stemming from distribution gaps between golden NAR tokens and synthetic NAR tokens. Therefore, applying preference optimization to calibrate the output distribution of NAR models is worth exploration.

9 Acknowledgements

This work was supported by the National Key Research and Development Program of China (No.2022ZD0160102). The computations in this research were performed using the CFFF platform of Fudan University.

References

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Teboul, O., Grangier, D., Tagliasacchi, M., and Zeghidour, N. Audiolm: a language modeling approach to audio generation, 2022.

Borsos, Z., Sharifi, M., Vincent, D., Kharitonov, E., Zeghidour, N., and Tagliasacchi, M. Soundstorm: Efficient parallel audio generation, 2023.

Budzianowski, P., Sereda, T., Cichy, T., and Vulić, I. Pheme: Efficient and conversational speech generation, 2024.

Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. Self-play fine-tuning converts weak language models to strong language models, 2024.

Cideron, G., Girgin, S., Verzetti, M., Vincent, D., Kastelic, M., Borsos, Z., McWilliams, B., Ungureanu, V., Bachem, O., Pietquin, O., Geist, M., Hussenot, L., Zeghidour, N., and Agostinelli, A. Musicrl: Aligning music generation to human preferences, 2024.

Défossez, A., Copet, J., Synnaeve, G., and Adi, Y. High fidelity neural audio compression, 2022.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Lee, K., Liu, H., Ryu, M., Watkins, O., Du, Y., Boutilier, C., Abbeel, P., Ghavamzadeh, M., and Gu, S. S. Aligning text-to-image models using human feedback, 2023.

Liao, H., Han, H., Yang, K., Du, T., Yang, R., Xu, Z., Xu, Q., Liu, J., Lu, J., and Li, X. Baton: Aligning text-to-audio model with human preference feedback, 2024.

Liu, H., Sferrazza, C., and Abbeel, P. Chain of hindsight aligns language models with feedback, 2023.

Liu, R., Sisman, B., and Li, H. Reinforcement learning for emotional text-to-speech synthesis with improved emotion discriminability, 2021.

OpenAI. Gpt-4 technical report, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision, 2022.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model, 2023.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. Learning to summarize from human feedback, 2022.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wang, C., Chen, S., Wu, Y., Zhang, Z., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., He, L., Zhao, S., and Wei, F. Neural codec language models are zero-shot text to speech synthesizers, 2023.

Wang, P., Zhang, D., Li, L., Tan, C., Wang, X., Ren, K., Jiang, B., and Qiu, X. Inferaligner: Inference-time alignment for harmlessness through cross-model guidance, 2024.

Xu, N., Zhao, J., Zu, C., Li, S., Chen, L., Zhang, Z., Zheng, R., Dou, S., Qin, W., Gui, T., Zhang, Q., and Huang, X. Advancing translation preference modeling with rlhf: A step towards cost-effective solution, 2024.

Yang, D., Tian, J., Tan, X., Huang, R., Liu, S., Chang, X., Shi, J., Zhao, S., Bian, J., Wu, X., Zhao, Z., Watanabe, S., and Meng, H. Uniaudio: An audio foundation model toward universal audio generation, 2023.

Yuan, H., Chen, Z., Ji, K., and Gu, Q. Self-play fine-tuning of diffusion models for text-to-image generation, 2024.

Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., and Tagliasacchi, M. Soundstream: An end-to-end neural audio codec, 2021.

Zhan, J., Dai, J., Ye, J., Zhou, Y., Zhang, D., Liu, Z., Zhang, X., Yuan, R., Zhang, G., Li, L., Yan, H., Fu, J., Gui, T., Sun, T., Jiang, Y., and Qiu, X. Anygpt: Unified multimodal llm with discrete sequence modeling, 2024.

Zhang, D., Li, S., Zhang, X., Zhan, J., Wang, P., Zhou, Y., and Qiu, X. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15757–15773, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.1055. URL `https://aclanthology.org/2023.findings-emnlp.1055`.

Zhang, D., Ye, R., Ko, T., Wang, M., and Zhou, Y. DUB: Discrete unit back-translation for speech translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 7147–7164, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.447. URL `https://aclanthology.org/2023.findings-acl.447`.

Zhang, D., Zhang, X., Zhan, J., Li, S., Zhou, Y., and Qiu, X. Speechgpt-gen: Scaling chain-of-information speech generation, 2024.

Zhang, T., Liu, F., Wong, J., Abbeel, P., and Gonzalez, J. E. The wisdom of hindsight makes language models better instruction followers, 2023c.

Zhang, X., Zhang, D., Li, S., Zhou, Y., and Qiu, X. Speechtokenizer: Unified speech tokenizer for speech large language models, 2023d.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide a detailed explanation of our contributions in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations and future work in Section **??**.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide detailed assumptions and complete proofs in Section 3 of the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a detailed description of the experimental setup and parameters.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: **[TODO]**

Guidelines: We'll open source the code once accepted.

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental details and settings are provided in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: This item is not necessary in our experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information about the computational resources used in the experiment in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential impact of our work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper does not have such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have used assets in accordance with the license and terms.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets introduced in the paper have well-documented records.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.