ACCELERATING DIFFUSION-BASED TEXT-TO-AUDIO GENERATION WITH CONSISTENCY DISTILLATION

Yatong Bai[*1,2] *Trung Dang*[2] *Dung Tran*[2] *Kazuhito Koishida*[2] *Somayeh Sojoudi*[1]

[1]University of California, Berkeley [2]Applied Sciences Group, Microsoft Corporation

ABSTRACT

Diffusion models power a vast majority of text-to-audio (TTA) generation methods. Unfortunately, these models suffer from slow inference speed due to iterative queries to the underlying denoising network, thus unsuitable for scenarios with inference time or computational constraints. This work leverages the recently proposed consistency models to distill TTA models that require only a single neural network query. In addition to moving the distillation process into the latent space and incorporating classifier-free guidance, we leverage the availability of generated audio during distillation training to fine-tune the consistency TTA model with novel loss functions in the audio space, such as the CLAP score. Our objective and subjective evaluation results on the AudioCaps dataset show that consistency models retain diffusion models' high generation quality and diversity while reducing the number of queries by a factor of 400.

***Index Terms*—** Diffusion models, Consistency models, Audio generation, Generative AI, Neural networks

1. INTRODUCTION

Text-to-audio (TTA) generation, which creates audio based on user-provided textual prompts, has recently gained significant popularity [1, 2, 3, 4, 5, 6, 7, 8, 9]. Many TTA models are based on latent diffusion models (LDM) [10] and have demonstrated the ability to produce diverse, precise, and high-quality audio. While LDMs are famous for their superior generation quality [11], they suffer from slow inference as they require iterative neural network queries, posing latency and computation challenges. As a result, accelerating diffusion-based TTA will make such technologies vastly more accessible, facilitating AI-assisted real-world art and media creation.

This work proposes to accelerate diffusion-based TTA with a novel *CFG-aware latent-space consistency model* that only requires a single neural network query per generation. Our approach moves the consistency model [12] into a latent space and incorporates classifier-free guidance (CFG) [13], which significantly enhances conditional generation, into the training process with three distinct approaches: direct guidance, fixed guidance, and variable guidance. To our knowledge, we are the first to introduce CFG to consistency models, not only for TTA but also for general content generation.

Moreover, leveraging the generated audio available during consistency model training, we propose to fine-tune the consistency TTA model end-to-end with prompt-aware audio quality objective functions. We use the CLAP score as an example objective to verify the enhanced sound quality and text correspondence. In contrast, diffusion models cannot take advantage of this process due to the prohibitive computation and the potential optimization difficulties arising from the recurrent inference process.

Our extensive experiments show that the consistency TTA model simultaneously achieves generation quality, speed, and diversity. Specifically, the generation quality of a single-network-query consistency model is comparable to a 400-query diffusion model across five objective metrics, as well as subjective audio quality and audio-text correspondence. We encourage the reader to listen to our generated examples at `consistency-tta.github.io/demo`.

When implemented with standard Python libraries, our consistency model takes an average of 9.1 seconds to generate one minute of audio on a laptop computer, whereas a representative diffusion method [1] needs more than a minute on a state-of-the-art A100 GPU for each minute-generation (details in Appendix A.7).

This paper is structured as follows. Section 2 reviews the related literature, Section 3 explains the technical innovations of our proposed TTA acceleration, and Section 4 discusses the experimental results. Additional discussions and details are presented in the appendix. Throughout this paper, vectors and matrices are denoted as bold symbols whereas scalars use regular symbols.

Shortly after the submission of this work, Luo et al. [14] applied our proposed CFG-aware latent-space consistency model to Stable Diffusion [11] and achieved exceptional quality-efficiency balance, confirming the efficacy of our methods for text-to-image. This concurrent work has since received attention from thousands, gaining multiple implementations in the image creation community, verifying our capability of making AI-assisted generation more accessible.

2. BACKGROUND AND RELATED WORK

2.1. Diffusion Models

Diffusion models [15, 16], known for their diverse, high-quality generations, have rapidly gained popularity among conditional and unconditional vision and audio generation tasks [11, 17, 3, 18]. In the vision domain, while pixel-level diffusion (e.g., EDM [17]) performs well on small image sizes, generating larger images usually requires LDMs [10], where the diffusion process takes place in a latent space. In the audio domain, generative tasks can be further categorized into speech, music, and in-the-wild audio creation. This paper considers the in-the-wild setting, where the goal is to generate diverse samples covering a variety of real-world sounds. While some works considered autoregressive models [8] or Mel-space diffusion [9], LDMs have emerged as the dominant TTA approach [1, 2, 3, 4, 5, 6, 7].

The intuition of diffusion models is to gradually recover a clean sample from a noisy sample. During training, Gaussian noise is progressively added to a ground-truth sample $\boldsymbol{z}_0$, forming a continuous diffusion trajectory. At the end of the trajectory, the noisy sample becomes indistinguishable from pure Gaussian noise. This trajectory is then discretized into N time steps, where the noisy sample at each step is denoted as $\boldsymbol{z}_n$ for $n = 1, \ldots, N$. In each training iteration, a random step n is selected, and a Gaussian noise with

*Work done during internship at Microsoft. Correspondence email `yatong_bai@berkeley.edu`.

variance depending on n is injected into the clean sample to produce $\boldsymbol{z}_n$. A denoising neural network, often a U-Net [19], is optimized to recover the noise distribution from the noisy sample. During inference, Gaussian noise is used to initialize the last noisy sample $\hat{\boldsymbol{z}}_N$, where $\hat{\boldsymbol{z}}_n$ denotes the predicted sample at step n. The diffusion model then generates a clean sample $\hat{\boldsymbol{z}}_0$ by iteratively querying the denoising network, producing the sequence $\hat{\boldsymbol{z}}_{N-1}, \ldots, \hat{\boldsymbol{z}}_0$.

2.2. Diffusion Inference Acceleration and Consistency Models

Diffusion models suffer from high generation latency and expensive inference computation due to iterative queries to the denoising network. Existing initiatives to reduce the number of model queries can mainly be grouped into improved samplers and distillation methods.

Improved samplers can reduce the number of inference steps N of existing diffusion models without additional training. Examples include DDIM [20], Euler [21], Heun, DPM [22, 23], PNDM [24], and Analytic-DPM [25]. The best samplers can reduce N to 10-50 from the hundreds required by vanilla inference using DDPM [16].

On the other hand, distillation methods, where a pre-trained diffusion model serves as the teacher and a student model is trained to mimic multiple teacher steps in a single step, can reduce the number of inference steps to below 10. One representative method is progressive distillation (PD) [26], which iteratively halves the number of steps. However, PD's single-step generation capability is still unideal, and the repetitive distillation procedure is time-consuming.

To this end, the consistency model [12] has been proposed for single-step fast generation without iterative distillation. The training goal of consistency models is to reconstruct the noiseless image within a single step from an arbitrary step on the diffusion trajectory.

Both PD and consistency models were proposed for image generation. Nonetheless, accelerating diffusion models in the audio domain is equally important if not more so, in order to enable interactive real-time audio generation with reasonable computation.

Previously, consistency models focused on pixel [12] or spectrogram-space [27] generation. Meanwhile, diffusion models demonstrated that latent-space generation greatly improves quality and details without excessively increasing the model size [11, 3, 1]. This work moves the consistency model into a latent space and demonstrates its effect in text-conditioned in-the-wild audio generation.

Consistency models were originally proposed for unconditional generation [12]. Conditional generation as in this work demands additional considerations, mainly CFG, which we discuss in Section 3.

2.3. Classifier-Free Guidance

CFG [13] is a highly effective method to adjust the conditioning strength for conditional generation models during inference. For diffusion models, CFG significantly enhances performance without additional training. Specifically, CFG obtains two noise estimations from the denoising network – one with conditioning (denoted as $\boldsymbol{v}_{\text{cond}}$) and one without (by masking the condition embedding, denoted as $\boldsymbol{v}_{\text{uncond}}$). The guided estimation $\boldsymbol{v}_{\text{cfg}}$ is obtained via

$$\boldsymbol{v}_{\text{cfg}} = w \cdot \boldsymbol{v}_{\text{cond}} + (1 - w) \cdot \boldsymbol{v}_{\text{uncond}}, \tag{1}$$

where the scalar $w \geq 0$ is the guidance strength. When w is between 0 and 1, CFG interpolates the conditioned and unconditioned estimations. When w is greater than 1, CFG becomes extrapolation.

Since CFG is external to the denoising network in diffusion models, it makes distillating guided models more complex than unguided ones. The authors of [28] outlined a two-stage pipeline for performing PD on a CFG model. It first absorbs CFG into the denoising network by letting the student network take w as an additional input, and then performs PD on the w-conditioned model. In both training stages, w is randomized, and the resulting distilled network allows for selecting w during inference.

3. CFG-AWARE LATENT-SPACE CONSISTENCY TTA

3.1. Overall Setup

We select TANGO [1], a state-of-the-art TTA framework, as the diffusion baseline. However, we highlight that most of the innovations in this paper also apply to other diffusion-based TTA models.

Similar to TANGO, our consistency model has four components: a conditional U-Net, a text encoder that processes the textual prompt, a VAE encoder-decoder pair that converts the Mel spectrogram to and from the U-Net latent space, and a HiFi-GAN vocoder [29] that produces time-domain audio waveform from the Mel spectrogram. We only train the U-Net and freeze other components.

During training, the Mel spectrogram of the audio is processed by the VAE encoder to produce a latent representation, and the prompt is transformed by the text encoder into a text embedding. They are given to the conditional U-Net as the input and the condition. The VAE decoder and the HiFi-GAN are not used.

During inference, the text embedding is used to guide the U-Net to reconstruct a latent audio representation. The Mel spectrogram and waveform are recovered by the VAE decoder and the HiFi-GAN vocoder, respectively. The VAE encoder is not used.

3.2. Consistency Distillation (CD)

The goal of CD is to learn a consistency student U-Net $f_{\text{S}}(\cdot)$ from the diffusion teacher module $f_{\text{T}}(\cdot)$ in TANGO. The neural architecture of f_{S} is the same as the f_{T}, taking three inputs: the noisy latent representation $\boldsymbol{z}_n$, the time step n, and the text embedding $\boldsymbol{e}_{\text{te}}$. Furthermore, the parameters in f_{S} are initialized using f_{T} information.

The goal for the student U-Net is to generate a realistic latent audio representation within a single forward pass, directly producing an estimated clean example $\hat{\boldsymbol{z}}_0$ based on $\boldsymbol{z}_n$, where $n \in \{0, \ldots, N\}$ is an arbitrary step along the diffusion trajectory [12, Algorithm 2]. The risk function to be minimized for achieving this goal is

$$\mathbb{E}_{\substack{(\boldsymbol{z}_0, \boldsymbol{e}_{\text{te}}) \sim \mathcal{D} \\ n \sim \text{U}_{\text{int}}(1, N)}} \Big[d\Big(f_{\text{S}}(\boldsymbol{z}_n, n, \boldsymbol{e}_{\text{te}}), f_{\text{S}}(\hat{\boldsymbol{z}}_{n-1}, n-1, \boldsymbol{e}_{\text{te}}) \Big) \Big], \tag{2}$$

where $d(\cdot, \cdot)$ is a distance measurement, $\mathcal{D}$ is the training dataset, $\text{U}_{\text{int}}(1, N)$ denotes the discrete uniform distribution supported over the set $\{1, \ldots, N\}$, and $\hat{\boldsymbol{z}}_{n-1} = \text{solve} \circ f_{\text{T}}(\boldsymbol{z}_n, n, \boldsymbol{e}_{\text{te}})$ is the teacher diffusion model's estimation for $\boldsymbol{z}_{n-1}$. Here, $\text{solve} \circ f_{\text{T}}$ denotes the composite function of the teacher denoising U-Net and the solver that converts the U-Net raw output to the estimation of the previous time step. We use the ℓ_2 distance in this latent space as $d(\cdot, \cdot)$, with additional discussions in Appendix A.6. Intuitively, this risk measures the expected distance between the student's reconstructions from two adjacent time steps on the diffusion trajectory.

The authors of [12] used the Heun solver to traverse the teacher model's diffusion trajectory during distillation and adopted "Karras noise schedule", a discretization scheme that unevenly selects the time steps on the diffusion trajectory. In Section 4.2, we empirically investigate multiple solvers and noise schedules.

3.3. Consistency Distillation with Classifier-Free Guidance

Since CFG is crucial to the conditional generation quality, we consider three methods for incorporating it into the distilled model.

Direct Guidance directly performs CFG on the consistency model output $\boldsymbol{z}_0$ by applying (1). Since this method naïvely extrapolates or interpolates the guided and unguided $\boldsymbol{z}_0$ predictions, it may move the prediction outside the manifold of realistic latent representations.

Fixed Guidance Distillation aims to distill from the diffusion model coupled with CFG using a fixed guidance strength w. Specifically, the training risk function is still (2), but $\hat{\boldsymbol{z}}_{n-1}$ is replaced with the estimation after CFG. Now, $\hat{\boldsymbol{z}}_{n-1}$ becomes $\text{solve} \circ f_{\text{T}}^{\text{cfg}}(\boldsymbol{z}_n, n, \boldsymbol{e}_{\text{te}}, w)$, where the guided teacher output $f_{\text{T}}^{\text{cfg}}$ is

$$f_{\text{T}}^{\text{cfg}}(\boldsymbol{z}_n, n, \boldsymbol{e}_{\text{te}}, w) = w \cdot f_{\text{T}}(\boldsymbol{z}_n, n, \varnothing) + (1-w) \cdot f_{\text{T}}(\boldsymbol{z}_n, n, \boldsymbol{e}_{\text{te}}),$$

with $\varnothing$ denoting the masked language token. Here, w is fixed to the value that optimizes teacher generation (3 is best for TANGO [1]).

Variable Guidance Distillation is similar to fixed guidance distillation, but with randomized guidance strength w during distillation, so that w can be adjusted during inference. To make the student network compatible with adjustable w, we add a w-encoding condition branch to f_{S} (which now has four inputs). We use Fourier encoding for w following [28] and merge the embedding into f_{S} similarly to the time step embedding. Each training iteration samples a random guidance strength w via the uniform distribution supported on $[0, 6)$.

The latter two methods are related to yet distinct from the two-stage distillation outlined in [28], with the details in Appendix A.5.

3.4. End-to-End Fine-Tuning with CLAP

Since our consistency TTA model produces audio in a single neural network query, we can optimize auxiliary losses operating in the audio space along with the latent-space CD loss to improve the audio quality and semantics. On the contrary, since a diffusion model has an iterative inference process, optimizing such a model by back-propagating from the audio resembles the training of a recurrent neural network, which is known to be expensive and challenging. This work uses the CLAP score [30] as an example of fine-tuning loss functions. The CLAP score, denoted by CS, is defined as:

$$\text{CS}(\hat{\boldsymbol{x}}, \boldsymbol{x}) = \max\left\{100 \times \frac{\mathbf{e}_{\hat{\boldsymbol{x}}} \cdot \mathbf{e}_{\boldsymbol{x}}}{\|\mathbf{e}_{\hat{\boldsymbol{x}}}\| \cdot \|\mathbf{e}_{\boldsymbol{x}}\|}, 0\right\}, \tag{3}$$

where $\hat{\boldsymbol{x}}$ is the generated audio waveform, $\boldsymbol{x}$ is the reference (ground-truth waveform or textual prompt), and $\mathbf{e}_{\hat{\boldsymbol{x}}}$ and $\mathbf{e}_{\boldsymbol{x}}$ are the corresponding embeddings extracted by the CLAP model.

We select the CLAP score due to its superior embedding quality arising from the diverse training tasks and datasets, as well as its consideration of audio-text correspondence. Since the CD objective (2) does not use ground truth information, optimizing this score provides valuable feedback to the consistency model.

3.5. Min-SNR Training Loss Weighting Strategy

The literature has proposed to improve diffusion models by using the signal-noise ratio (SNR) to weigh the training loss at each time step n, and Min-SNR [31] is one of the latest strategies. The Min-SNR calculation depends on whether the diffusion model predicts the clean example $\boldsymbol{z}_0$, the additive noise $\boldsymbol{\epsilon}$, or the noise velocity $\boldsymbol{v}$.

This work investigates how Min-SNR affects CD. Since consistency models predict the clean sample $\boldsymbol{z}_0$, we use the Min-SNR formulation for $\boldsymbol{z}_0$-predicting diffusion models, which is $\omega(n) = \min\{\text{SNR}(t_n), \gamma\}$, where $\omega(n)$ is the loss weight for the n^{th} time step, $\text{SNR}(t)$ is the SNR at time t, t_n is the time corresponding to the n^{th} time step, and γ is a constant defaulted to 5. For the Heun solver used in most of our experiments, $\text{SNR}(t)$ is the inverse of the additive Gaussian noise variance at time t.

4. EXPERIMENTS

4.1. Dataset and Experiment Settings

The evaluation of our models uses AudioCaps [32], a popular in-the-wild audio dataset that TTA methods regard as the go-to benchmark [1, 2, 3, 8]. AudioCaps is a collection of human-captioned YouTube audio, each instance at most ten seconds long. Our AudioCaps copy contains 45,260 training examples, and we use the test subset from [1] with 882 instances. Like several existing works [1, 3], the core U-Net of our models is trained only on the AudioCaps training set, demonstrating high data efficiency. Pre-training on larger datasets may further improve our results, which we leave for future work.

While we explicitly use TANGO [1] as the diffusion baseline, our methods apply to diffusion-based TTA models in general. We select FLAN-T5-Large [33] as the text encoder and use the same checkpoint as [1]. For the VAE and the HiFi-GAN, we use the checkpoint pre-trained on AudioSet released by the authors of [3] as in [1]. For faster training and inference, we shrink the U-Net from 866M parameters used in [1] to 557M. As shown in Table 2, this smaller TANGO model performs similarly to the checkpoint from [1]. All consistency models are subsequently distilled from this smaller model. Additional details about our model, training setup, and evaluation are shown in Appendix A.6 and A.7.

4.2. Objective Evaluation Results

Our objective evaluation considers five metrics: FAD, FD, KLD, CLAP_{A}, and CLAP_{T}. The former four use the ground-truth audio as the reference, whereas CLAP_{T} uses the text. Specifically, FAD is the Fréchet distance between generated and ground-truth audio embeddings extracted by VGGish [34], whereas FD and KLD are the Fréchet distance and the Kullback-Leibler divergence between the PANN [35] audio embeddings. CLAP_{A} and CLAP_{T} are the CLAP scores with respect to the ground-truth audio and the textual prompt.

We first ablate the performance of our consistency TTA model under various distillation settings, presenting the results in Table 1. All runs use $N = 18$ diffusion discretization steps during distillation as in [12]. Table 1 demonstrates that distilling with fixed or variable guidance significantly improves all metrics over direct or no guidance, highlighting the importance of CFG-aware distillation. While $w = 3$ is the optimal CFG weight for the teacher diffusion model, the optimal w becomes larger for the consistency model obtained with variable guidance distillation, aligning with the observations in [28]. Meanwhile, using the more accurate Heun solver to traverse the teacher model's diffusion trajectory for distillation is more advantageous than using the simpler DDIM solver. Somewhat surprisingly, the uniform noise schedule is preferred over the Karras schedule, as the former results in superior consistency model FAD, FD, and KLD (see Appendix A.2 for more detailed discussions). We also observe that Min-SNR loss weighting and guided initialization improve the FD and FAD but slightly sacrifice the KLD. Here, "guided initialization" refers to initializing the consistency model with a CFG-aware diffusion model, whereas "unguided initialization" refers to initializing with the unmodified TANGO teacher weights.

On top of the best consistency distillation setting, we perform end-to-end CLAP fine-tuning, co-optimizing three loss components: the consistency loss (2), CLAP_{A}, and CLAP_{T}. Table 2 shows that fine-tuning improves not only CLAP scores but also FAD and FD.

Table 2 additionally compares our consistency TTA models with several state-of-the-art diffusion baseline TTA models. When using the best CFG weight, the audio quality of the consistency model is similar to that of the diffusion models: all objective metrics are

Table 1. Ablate various guidance weights, distillation techniques, solvers, noise schedules, training lengths, loss weights, and initialization. "CFG w" represents the guidance weight; "# queries" indicates the number of neural network queries during inference. U-Net modules have 557M parameters, except in variable guidance models (559M). Distillation runs are 40 epochs; inference uses FP32 precision.

Solver	Noise Schedule	CFG w	Guidance Method	Min-SNR	Initialization	# Queries (↓)	FAD (↓)	FD (↓)	KLD (↓)
DDIM	Uniform	1	-	✗	Unguided	1	13.48	45.75	2.409
Heun	Karras						10.97	50.19	2.425
DDIM	Uniform	3	Direct Guidance	✗	Unguided	2	8.565	38.67	2.015
Heun	Karras						7.421	39.36	1.976
Heun	Karras	3	Fixed Guidance Distillation	✗	Unguided	1	5.702	33.18	1.494
	Uniform			✗	Unguided		4.168	28.54	1.384
	Uniform			✓	Unguided		3.766	27.74	1.443
	Uniform			✗	Guided		3.859	27.79	1.421
Heun	Uniform	3	Variable Guidance Distillation	✓	Guided	1	3.956	28.27	1.442
		4					3.180	27.92	1.394
		6					2.975	28.63	1.378

Table 2. Compare consistency models to the diffusion baselines. Distillation runs are extended to 60 epochs for better performance; CLAP-fine-tuning uses 10 additional epochs. All CD runs use the Heun teacher solver, uniform noise schedule, variable guidance distillation, guided initialization, Min-SNR loss weights, and BF16 inference precision. Bold numbers indicate the best results among consistency models.

	U-Net # Params	CFG w	# Queries (↓)	CLAP_T (↑)	CLAP_A (↑)	FAD (↓)	FD (↓)	KLD (↓)
AudioLDM-L reported in [3]	739M	2	400	-	-	2.08	27.12	1.86
TANGO reported in [1]	866M	3		-	-	1.59	24.53	1.37
TANGO [1] tested by us	866M	3		24.10	72.85	1.631	20.11	1.362
Our TANGO model	557M	3		24.57	72.79	1.908	19.57	1.350
Consistency model (ours) without CLAP fine-tuning	559M	3	1	21.00	71.39	3.202	22.04	1.411
		4		22.05	72.08	2.610	21.71	1.373
		5		22.50	72.30	2.575	22.08	**1.354**
Consistency model (ours) with CLAP fine-tuning	559M	3	1	24.44	72.39	**2.182**	**20.44**	1.368
		4		24.69	**72.54**	2.406	20.97	1.358
		5		**24.70**	72.53	2.626	21.33	1.356

Table 3. Compare the human evaluation results of consistency and diffusion models. Bold numbers are defined same as Table 2.

	U-Net # Params	CLAP Fine-Tuning	CFG w	# Queries (↓)	Human Quality (↑)	Human Corresp (↑)	CLAP_T (↑)	CLAP_A (↑)	FAD (↓)	FD (↓)	KLD (↓)
Diffusion	557M	✗	3	400	4.136	4.064	24.57	72.79	1.908	19.57	1.350
Consistency (ours)	559M	✗	5	1	**3.902**	4.010	22.50	72.30	2.575	22.08	**1.354**
		✓	4		3.830	**4.064**	**24.69**	**72.54**	**2.406**	**20.97**	1.358
Ground-Truth	-	-	-	-	4.424	4.352	26.71	100.0	0.000	0.000	0.000

close, with the consistency TTA's FD and KLD even better than the reported numbers in [1] and [3]. Note that the diffusion models use 200 inference steps following [3, 1], each step requiring two noise estimations due to CFG, summing to 400 network queries per generation. In other words, with minimal performance drop, the proposed method reduces the number of U-Net queries by a factor of 400.

In Appendix A.1, we compare consistency TTA models with training-free acceleration methods listed in Section 2.2, demonstrating that our method generates better audio faster than existing works. In Appendix A.3, we discuss the significant real-world computing time reduction of consistency TTA models.

4.3. Subjective Evaluation Results

Finally, we conduct subjective evaluations in two aspects: overall audio quality and audio-text correspondence. For each subject, we use 25 generated audio clips from the same set of prompts together with those from ground-truth samples. We instructed 20 evaluators to rate the audio clips on a scale of 1 to 5 for each aspect. Other details can be found in Appendix A.7. The subjective scores presented in Table 3 further confirm that the consistency model produces audios close to those of the diffusion model in terms of subjective evaluation scores. Moreover, optimizing the CLAP scores improves the text-audio correspondence score, which supports our assumption that CLAP_T provides closed-loop feedback to help align the generated audio with the prompt.

4.4. Diversity of Generated Audio

We also observe that different random seeds, i.e., different initial Gaussian latent for the consistency TTA model, generate noticeably different audio, confirming that consistency models produce diverse generations like diffusion models. To demonstrate this, we present the generated audio from the first 50 AudioCaps test prompts with four different seeds at `consistency-tta.github.io/diversity`, and display corresponding spectrogram examples in Appendix A.4, where we also provide quantitative evidence.

5. CONCLUSION

This work proposes a novel approach to accelerate the core module of latent-diffusion-based TTA models hundreds of times based on consistency models, making AI-assisted audio generation more efficient and accessible than ever, for AI experts, audio professionals, and hobbyists alike. The delicate distillation procedure that emphasizes CFG achieves this vast speed-up with minimal generation quality reduction, enabling diverse and realistic in-the-wild audio generation within one neural network query. The differentiability of the resulting model allows for end-to-end fine-tuning, unlocking possibilities for revolutionizing the training method of generative models.

6. REFERENCES

[1] Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria, "Text-to-audio generation using instruction-tuned llm and latent diffusion model," *arXiv preprint arXiv:2304.13731*, 2023.

[2] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," *Transactions on Audio, Speech, and Language Processing*, 2023.

[3] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," *arXiv preprint arXiv:2301.12503*, 2023.

[4] Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley, "AudioLDM 2: Learning holistic audio generation with self-supervised pretraining," *arXiv preprint arXiv:2308.05734*, 2023.

[5] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao, "Make-an-Audio: Text-to-audio generation with prompt-enhanced diffusion models," *arXiv preprint arXiv:2301.12661*, 2023.

[6] Jiawei Huang, Yi Ren, Rongjie Huang, Dongchao Yang, Zhenhui Ye, Chen Zhang, Jinglin Liu, Xiang Yin, Zejun Ma, and Zhou Zhao, "Make-an-Audio 2: Temporal-enhanced text-to-audio generation," *arXiv preprint arXiv:2305.18474*, 2023.

[7] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal, "Any-to-any generation via composable diffusion," *arXiv preprint arXiv:2305.11846*, 2023.

[8] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi, "AudioGen: Textually guided audio generation," in *International Conference on Learning Representations*, 2023.

[9] Seth Forsgren and Hayk Martiros, "Riffusion - stable diffusion for real-time music generation," *URL https://riffusion.com*, 2022.

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, "High-resolution image synthesis with latent diffusion models," in *Conference on Computer Vision and Pattern Recognition*, 2022.

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, "High-resolution image synthesis with latent diffusion models," in *Conference on Computer Vision and Pattern Recognition*, 2022.

[12] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever, "Consistency models," in *International Conference on Machine Learning*, 2023.

[13] Jonathan Ho and Tim Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[14] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao, "Latent consistency models: Synthesizing high-resolution images with few-step inference," *arXiv preprint arXiv:2310.04378*, 2023.

[15] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, 2015.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, 2020.

[17] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine, "Elucidating the design space of diffusion-based generative models," in *Advances in Neural Information Processing Systems*, 2022.

[18] Qingqing Huang, Daniel S Park, Tao Wang, Timo I Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, et al., "Noise2Music: Text-conditioned music generation with diffusion models," *arXiv preprint arXiv:2302.03917*, 2023.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015.

[20] Jiaming Song, Chenlin Meng, and Stefano Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[21] Leonhard Euler, *Institutionum calculi integralis*, vol. 1, impensis Academiae imperialis scientiarum, 1824.

[22] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu, "DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps," in *Advances in Neural Information Processing Systems*, 2022.

[23] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu, "DPM-solver++: Fast solver for guided sampling of diffusion probabilistic models," *arXiv preprint arXiv:2211.01095*, 2022.

[24] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao, "Pseudo numerical methods for diffusion models on manifolds," in *International Conference on Learning Representations*, 2022.

[25] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang, "Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models," in *International Conference on Learning Representations*, 2021.

[26] Tim Salimans and Jonathan Ho, "Progressive distillation for fast sampling of diffusion models," in *International Conference on Learning Representations*, 2021.

[27] Zhen Ye, Wei Xue, Xu Tan, Jie Chen, Qifeng Liu, and Yike Guo, "CoMoSpeech: One-step speech and singing voice synthesis via consistency model," *arXiv preprint arXiv:2305.06908*, 2023.

[28] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans, "On distillation of guided diffusion models," in *Conference on Computer Vision and Pattern Recognition*, 2023.

[29] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, 2020.

[30] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang, "CLAP: learning audio concepts from natural language supervision," in *International Conference on Acoustics, Speech and Signal Processing*, 2023.

[31] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo, "Efficient diffusion training via min-snr weighting strategy," *arXiv preprint arXiv:2303.09556*, 2023.

[32] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim, "AudioCaps: Generating captions for audios in the wild," in *Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[33] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al., "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022.

[34] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al., "CNN architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing*, 2017.

[35] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[36] Rongjie Huang, Max WY Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao, "Fastdiff: A fast conditional diffusion model for high-quality speech synthesis," in *International Joint Conference on Artificial Intelligence*, 2022.

[37] Brian McFee, "ResamPy: efficient sample rate conversion in python," *Journal of Open Source Software*, vol. 1, no. 8, pp. 125, 2016.

[38] Yao-Yuan Yang, Moto Hira, Zhaoheng Ni, Anjali Chourdia, Artyom Astafurov, Caroline Chen, Ching-Feng Yeh, Christian Puhrsch, David Pollack, Dmitriy Genzel, Donny Greenberg, Edward Z. Yang, Jason Lian, Jay Mahadeokar, Jeff Hwang, Ji Chen, Peter Goldsborough, Prabhat Roy, Sean Narenthiran, Shinji Watanabe, Soumith Chintala, Vincent Quenneville-Bélair, and Yangyang Shi, "TorchAudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.

[39] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *International Conference on Acoustics, Speech and Signal Processing*, 2023.

[40] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *International Conference on Acoustics, Speech and Signal Processing*, 2017.

Table 4. Compare our consistency TTA model with training-free diffusion acceleration methods, specifically improved ODE solvers. All diffusion models use the same TANGO weights as in Table 3 and use a CFG weight of $w = 3$. All solvers use the uniform noise schedule, except for "Heun+Karras", which uses the noise schedule proposed in [17] with the Heun solver.

Model Type	Solver	# Queries (↓)	FAD (↓)	FD (↓)	KLD (↓)
Diffusion (default 200 steps)	DDPM	400	1.908	19.57	1.350
Diffusion (8 steps)	DDPM	16	17.29	56.23	1.897
	DDIM	16	9.859	32.45	1.432
	Euler	16	7.693	35.42	1.452
	DPM++(2S)	32	2.543	25.29	1.350
	Heun	32	2.481	24.65	1.377
	Heun+Karras	32	2.721	26.43	1.398
Diffusion (5 steps)	Heun	20	5.729	30.05	1.495
Consistency (ours, 1 step)	-	1	2.575	22.08	1.354

A. ADDITIONAL DISCUSSIONS AND DETAILS

A.1. Comparison with Training-Free Acceleration Methods

This section compares consistency models with diffusion acceleration methods that do not require tuning the model weights. As mentioned in Section 2.2, most training-free acceleration methods focus on improved sampling strategies, aiming to use the noise estimation from the denoising network more efficiently. While these methods can effectively reduce the number of denoising queries while mostly maintaining generation quality, they struggle to bring the inference steps below 5-15, and each step may require multiple denoising queries due to CFG and high solver order. In Table 4, we compare our single-step consistency models with training-free methods.

As shown in Table 4, with the help of improved ordinary differential equation (ODE) solvers, when the number of inference steps is reduced to 8 from the default setting of 200, the diffusion model can still generate audio with reasonable quality. Among these solvers, Heun achieves the best generation quality. Since Heun is a second-order solver that requires two noise estimations per step and each noise estimation requires two model queries due to CFG, 8-step inference with the Heun solver requires 32 model queries, demanding significantly more computation than our consistency model while achieving worse objective generation quality. Moreover, if we attempt to further reduce the number of inference steps from 8 to 5, the resulting audio noticeably deteriorates even with the Heun solver.

In addition those presented in Table 4, other training-free acceleration methods include Analytic-DPM [25] and FastDiff [36]. Analytic-DPM is an older work from the team that devised the DPM and DPM++ solvers [22, 23], with the latter included in Table 4. The authors of [22] demonstrated that DPM-solver achieves better generation quality than Analytic-DPM within even fewer steps, and DPM++ further improves (DPM and DPM++ solvers are also much more popular and easier to implement). Meanwhile, FastDiff makes architectural changes to tailor text-to-speech. Therefore, it requires training a new model and is difficult to integrate without significant modifications. Note that both Analytic-DPM and FastDiff are still few-step methods, which are much slower than our single-query consistency model. On the other hand, previous distillation methods such as PD [26] require prohibitively expensive training.

A.2. Additional Discussions Regarding the Teacher Solver

Table 1 presents the generation quality of the consistency model f_{S} distilled with various solver settings, confirming our selection of the Heun solver. This result aligns with the observations of [12]. Moreover, as shown in Table 4, among all experimented solvers, Heun optimizes the teacher diffusion model's generation quality for a fixed number of inference steps, further supporting our usage of the Heun solver for harnessing the teacher model during consistency distillation.

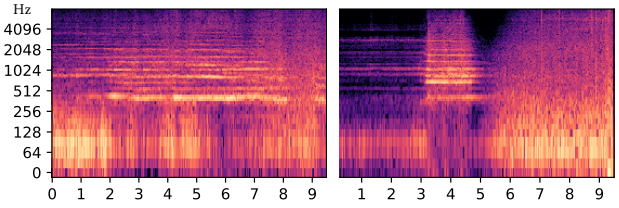
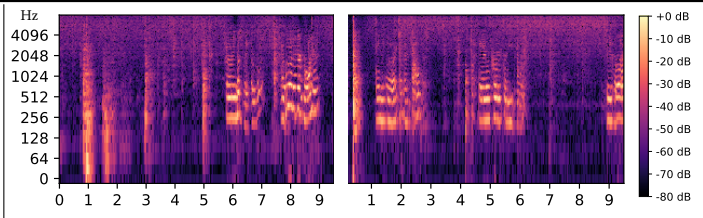
Intuitively, using the more delicate Heun solver is beneficial because it allows the distillation process to follow the diffusion trajectory accurately without discretizing the diffusion trajectory into a large number of steps (i.e., use a very large N). Using a large N during CD is undesirable because adjacent discretization steps will be very close. Since the training objective of consistency models is to minimize the difference between the predicted noiseless samples from adjacent points on the diffusion trajectory, a fine-grained discretization implies that each training step only provides very little information. Thus, a smaller N paired with an accurate ODE solver such as Heun is more suitable.

On the other hand, Table 1 also suggests that the uniform noise schedule is preferred over the Karras schedule (DDIM+uniform $\approx$ Heun+Karras $<$ Heun+uniform). This observation is surprising because the authors of [12] suggested using the Karras schedule. Our explanation for this difference is that TANGO was trained using the uniform schedule, whereas the teacher models in [12] were trained with the Karras schedule. It is likely beneficial for the distillation procedure to adopt the same noise schedule as the diffusion teacher training.

A.3. Inference Computing Time Comparison

On an Nvidia A100 GPU, generating from all 882 AudioCaps test prompts requires 2.3 minutes with our consistency model. The default TANGO model needs 168 minutes (73 minutes with the smaller 557M U-Net), 72 times as long compared with our consistency model. Note that the 200-step default inference schedule is shared among multiple diffusion-based TTA methods [1, 3], and thus, this TANGO inference time is representative. Moreover, our consistency model can run on a standard laptop computer, only taking 76 seconds to generate 50 ten-second audio clips, averaging 9.1 seconds per minute-generation. In contrast, the default TANGO requires 68 seconds per minute-generation on a state-of-the-art A100 GPU.

Table 5. The generated audio noticeably varies with different random seeds. The horizontal axis is time in seconds.

Seed 0	Seed 20230817	Seed 0	Seed 20230817
			
A train sounds horn and travels.		Food sizzling with some knocking and banging followed by a woman speaking.	

We note that the computing time depends on many software and hardware settings, with different model types affected to different degrees, and therefore these results are only for reference. Specifically, the above results are timed with off-the-shelf PyTorch code, and real-world speed-up can be even more prominent with implementation optimizations, approaching the hundreds-fold theoretic acceleration.

A.4. More Evidences of Generation Diversity

The generation diversity of consistency models is inherent due to its connection to diffusion models. Since consistency models operate on the diffusion trajectories as do diffusion models, their generations from the same initial noise should be similar (as shown in Figures 5 and 15 of [12]). As a result, the generation diversity of consistency models is on par with diffusion models, which are known to be highly diverse.

This section presents the generated spectrograms from the consistency models using different seeds, demonstrating that consistency TTA models simultaneously achieve efficient generation and diversity, a goal previous models struggled to reach. Table 5 presents the generated spectrograms (calculated via performing STFT on the generated waveforms) from two example prompts with two different seeds, whereas Figure 1 presents the Mel spectrograms (VAE decoder outputs before the vocoder) of the first 50 AudioCaps test prompts generated with four different seeds (corresponding to the audio examples on `consistency-tta.github.io/diversity`). It is apparent that the generations from the same prompt with different seeds are correlated but distinctly different.

For quantitative evidence, we collect the Mel spectrograms of these 50 generations across four seeds, standardize them individually, calculate the standard deviation across different seeds, and average the deviations across all Mel spectrogram points of the 50 prompts. The average number is 0.871, again demonstrating non-trivial generation diversity.

Another quantitative metric that considers diversity is the Inception Score (IS). Note that IS (higher is better) measures the diversity from an alternative perspective – across different prompts rather than different seeds, while also accounting for audio quality. As in [3], we use the PANN model embeddings for IS calculation. Our consistency TTA models reach an IS of 8.29/8.88 before/after CLAP fine-tuning, surpassing AudioLDM [3], which reported 8.13, and TANGO [1], which achieved 8.26 (test by us since [1] did not report IS).

A.5. Relationship to Two-Stage Progressive Distillation

Unlike PD in [28], which requires iteratively halving the number of diffusion steps, CD in our method reduces the required inference step to one within a single training process. As a result, the two distillation stages proposed in [28] can be merged. Specifically, Stage-2 distillation can be performed without Stage 1, provided that the step of querying the stage-1 model is replaced by querying the original teacher model with CFG. Merging Stage 1 and Stage 2 then results in our "variable guidance distillation" method discussed in Section 3.3. Subsequently, Stage 1 becomes optional since it only serves to provide a guidance-aware initialization to Stage 2.

A.6. Model and Training Details

We noticed that the audio resampling implementation has a major influence on some metrics, with FAD being especially sensitive. To ensure high training quality and fair evaluation, we use ResamPy [37] for all resampling procedures unless the resampling step needs to be differentiable. Specifically, CLAP fine-tuning requires differentiable resampling, and we use TorchAudio [38] instead.

The structure of our 557M-parameter U-Net is similar to the 866M U-Net used in [1], with the only modification being reducing the "block out channels" from $(320, 640, 1280, 1280)$ to $(256, 512, 1024, 1024)$. All CD runs use two 48GB-VRAM GPUs, with a total batch size of 12 and five gradient accumulation steps. The optimizer is AdamW with a 10^{-4} weight decay, and the learning rate is 10^{-5} for CD and 10^{-6} for CLAP fine-tuning. The "CD target network" (see [12] for details) is an exponential model average (EMA) copy with a 0.95 decay rate. We also maintain an EMA copy with a 0.999 decay rate for the reported experiment results. All training uses BF16 numerical precision.

Regarding the distance measure $d(\cdot, \cdot)$ introduced in (2), the authors of [12] considered several options for $d(\cdot, \cdot)$ for image generation tasks and concluded that using LPIPS (an evaluation metric that embeds the generated image with a deep model and calculates the weighted distance in several feature spaces) as the optimization objective produced higher generation quality than using the pixel-level ℓ_2 or ℓ_1 distance. However, since our latent diffusion model already operates in a latent feature space, we simply use the ℓ_2 distance in this latent space.

A.7. Evaluation Details

While the maximal audio length of the AudioCaps dataset is 10.00 seconds and the U-Net module of the TTA models is trained to generate 10.00-second latent audio representations, the HiFi-GAN vocoder produces 10.24-second audio. We observe that this mismatch negatively impacts the generation quality. Specifically, the final 0.24 seconds of the generated audio is empty, and there are slight distorting artifacts near the end of the 10-second useful portion. To this end, for the objective evaluation results in Tables 2 and 3, we truncate the generated audio to 9.70 seconds. Table 1 uses the full 10.24 seconds. The ground-truth reference waveforms are kept as-is. For CLAP_A and CLAP_T calculations, we use the CLAP checkpoint from [39] trained on LAION-Audio-630k [39], AudioSet [40], and music.

The human evaluation results in Table 3 are based on 20 evaluators each rating 25 audio clips per model, forming 500 samples per model. For each evaluator, the three models and the ground truth use the same set of prompts (the prompts vary across evaluators). Each evaluator rates each audio on a scale of 1-5, with rating criteria defined in the evaluation form. To ensure evaluation fairness, the model type generating each waveform is not disclosed to the evaluator, and the generations of the models are shuffled. We find it extremely challenging for a human to distinguish the outputs from the three generative models, with many ground truth waveforms also indistinguishable. An example evaluation form is available at `consistency-tta.github.io/evaluation`.

B. ACKNOWLEDGMENTS

We sincerely appreciate the contributions to human evaluation from Chih-Yu Lai, Mo Zhou, Afrina Tabassum, You Zhang, Sara Abdali, Uros Batricevic, Yinheng Li, Asing Chang, Rogerio Bonatti, Sam Pfrommer, Ziye Ma, Tanvir Mahmud, Eli Brock, Tanmay Gautam, Jingqi Li, Brendon Anderson, Hyunin Lee, and Saeed Amizadeh.

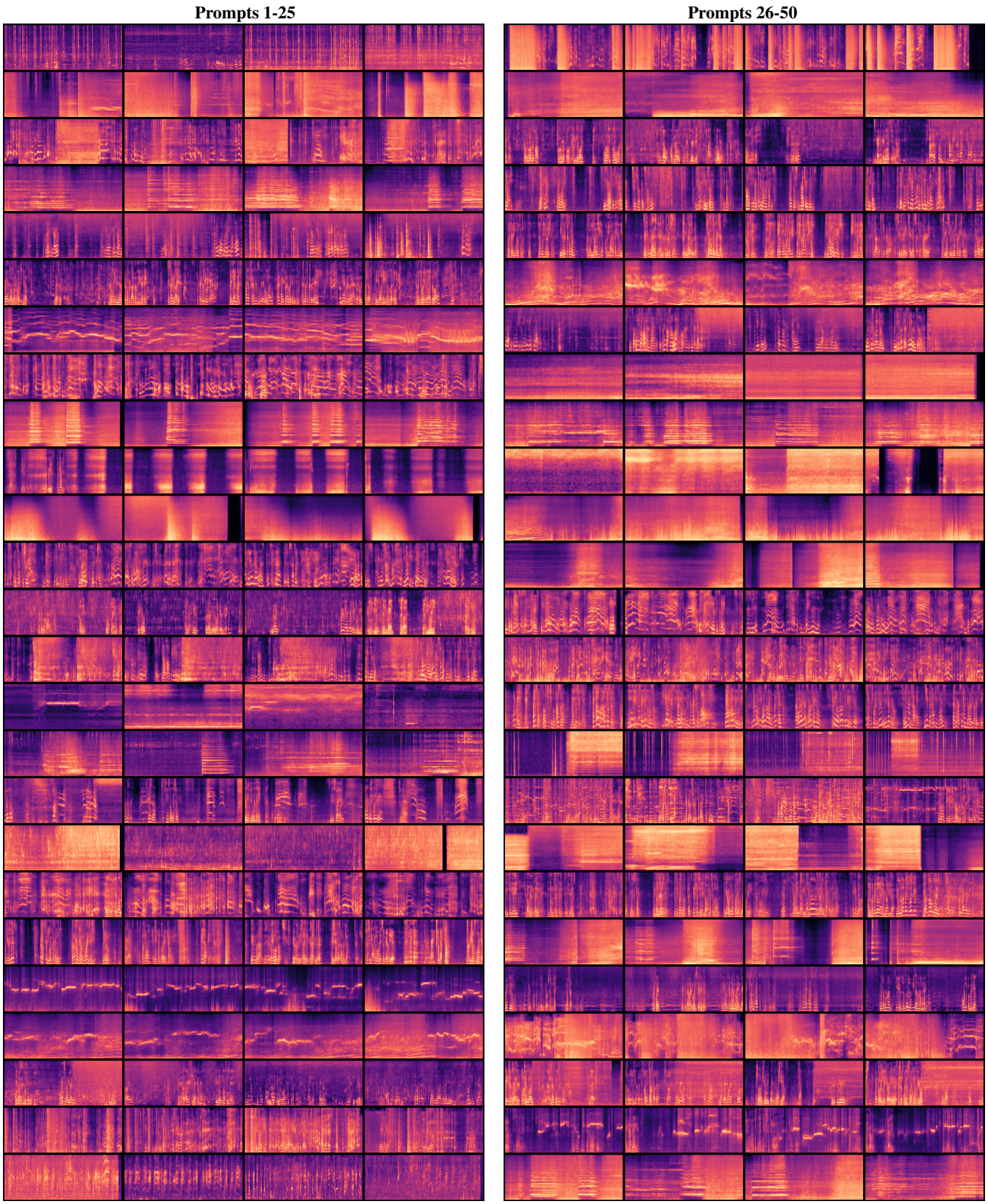


Fig. 1. Consistency model generated Mel spectrograms from the first 50 AudioCaps prompts with four different seeds. Each row corresponds to a prompt, and each column corresponds to a seed. The generations from a prompt with different seeds are correlated but distinctly different.