

# ConsistencyTTA: Accelerating Diffusion-Based Text-to-Audio Generation with Consistency Distillation

Anonymous submission to Interspeech 2024

## Abstract

Diffusion models are instrumental in text-to-audio (TTA) generation. Unfortunately, they suffer from slow inference due to an excessive number of queries to the underlying denoising network per generation. To address this bottleneck, we introduce ConsistencyTTA, a framework requiring only a single non-autoregressive network query, thereby accelerating TTA by hundreds of times. We achieve so by proposing “CFG-aware latent consistency model,” which adapts consistency generation into a latent space and incorporates classifier-free guidance (CFG) into model training. Moreover, unlike diffusion models, ConsistencyTTA can be finetuned closed-loop with audio-space text-aware metrics, such as CLAP score, to further enhance the generations. Our objective and subjective evaluation on the AudioCaps dataset shows that compared to diffusion-based counterparts, ConsistencyTTA reduces inference computation by 400x while retaining generation quality and diversity.

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

## 1. Introduction

Text-to-audio (TTA) generation, which synthesizes diverse auditory content from textual prompts, has garnered substantial interest within the scientific community [1, 2, 3, 4, 5, 6, 7, 8, 9]. Instrumental to this advancement are latent diffusion models (LDM) [10], which are famous for superior generation quality and diversity [10]. Unfortunately, LDMs suffer from prohibitively slow inference as they require excessive iterative neural network queries, posing considerable latency and computation challenges. Hence, accelerating diffusion-based TTA can greatly broaden their use and lower their environmental impact, making AI-driven media creation more feasible in practice.

We propose *ConsistencyTTA*, which accelerates diffusion-based TTA hundreds of times with negligible generation quality and diversity degradation. Central in our approach are two innovations: (1) a novel *CFG-aware latent-space consistency model* requiring only a single non-autoregressive network query per generation and (2) *closed-loop finetuning with audio-space text-aware metrics*. More specifically, ConsistencyTTA adapts consistency model [11] into a latent space and incorporates classifier-free guidance (CFG) [12] into training to significantly enhance conditional generation quality. We analyze three approaches for CFG: direct guidance, fixed guidance, and variable guidance. To our knowledge, we are the first to introduce CFG into CMs, for both TTA and general content generation.

Additional results, details, and discussions presented in the supplemental material [consistency-tta.github.io/report.pdf](https://consistency-tta.github.io/report.pdf).

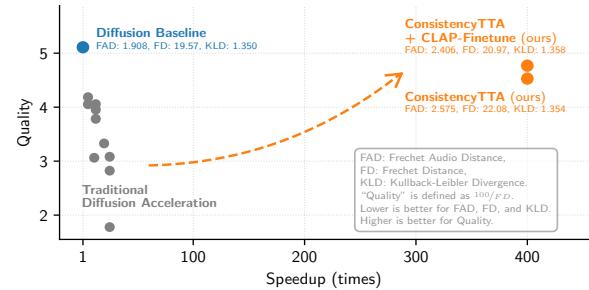


Figure 1: *ConsistencyTTA* achieves a 400x computation reduction compared with a diffusion baseline model while sacrificing much less quality than traditional acceleration methods.

Moreover, a distinct advantage of consistency models (CM) is the availability of generated audio during training, unlike diffusion models, whose generations are inaccessible during this phase due to their recurrent inference process. This allows closed-loop finetuning ConsistencyTTA with audio quality and audio-text correspondence objectives to further enhance generation quality. We use CLAP [13] as an example objective and verify the improved generation quality and text correspondence.

We focus on in-the-wild audio generation which produces a wide array of samples capturing the diversity of real-world sounds. Our extensive experiments, summarized in Figure 1, show that ConsistencyTTA achieves high-quality, fast, and diverse generations simultaneously. Specifically, the generation quality of the single-network-query ConsistencyTTA is comparable to a 400-query diffusion model across five objective metrics and two subjective metrics (audio quality and audio-text correspondence). The raw data and explanations of Figure 1 are provided in Appendix A.1. Our demo is located at the anonymous website<sup>1</sup>.

Using standard PyTorch implementation, ConsistencyTTA generates one minute of audio in just 9.1 seconds on a laptop computer. In contrast, a representative diffusion method [1] requires over a minute on a state-of-the-art A100 GPU for an equivalent task (details in Appendix B.5).

## 2. Background and Related Work

Throughout this paper, vectors and matrices are denoted as bold symbols, while scalars use regular symbols.

### 2.1. Diffusion Models

Diffusion models [14, 15], known for their diverse and high-quality generations, have rapidly gained popularity across vi-

<sup>1</sup>[consistency-tta.github.io/demo-anony](https://consistency-tta.github.io/demo-anony)

sion and audio generation tasks [10, 16, 3, 17, 18]. In vision, while pixel-level diffusion (e.g., EDM [16]) excel in generating small images, producing larger images requires LDMs [10] as they facilitate the diffusion process within a latent space. In the audio domain, 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, isotropic Gaussian noise is progressively added to a ground-truth sample  $z_0$ , forming a continuous diffusion trajectory. At the end of the trajectory, the noisy sample becomes indistinguishable from pure Gaussian noise. Discretizing the trajectory into  $N$  time steps and denoting the noisy sample at each step as  $z_n$  for  $n = 1, \dots, N$ , each training iteration selects a random step  $n$  and injects Gaussian noise, whose variance depends on  $n$ , into the clean sample to produce  $z_n$ . A denoising neural network, often a U-Net [19], is optimized to estimate the added noise from the noisy sample. During inference, Gaussian noise is used to initialize the last noisy sample  $\hat{z}_N$ , where  $\hat{z}_n$  denotes the predicted sample at step  $n = 1, \dots, N$ . The diffusion model then generates a clean sample  $\hat{z}_0$  by iteratively querying the denoising network, producing the sequence  $\hat{z}_{N-1}, \dots, \hat{z}_0$ .

## 2.2. Diffusion Acceleration and Consistency Models

Despite their high-quality generations, diffusion models suffer from prohibitive latency and costly inference computation due to iterative queries to the denoising network. Initiatives to reduce the model query number include improved samplers (training-free) and distillation methods (training-based).

Improved samplers, such as DDIM [20], Euler [21], Heun, DPM [22, 23], PNDM [24], and Analytic-DPM [25], reduce the number of inference steps  $N$  of trained diffusion models without additional training. The best samplers can reduce  $N$  from the hundreds required by vanilla DDPM [15] to 10-50. However, reducing  $N$  to below 10 remains a major challenge. Conversely, distillation methods, wherein a pre-trained diffusion model acts as the ‘teacher’ and a ‘student’ model is subsequently trained to emulate several teacher steps in a single step, can reduce the number of inference steps below 10 [26, 11, 27]. Progressive distillation (PD) [26] exemplifies such a method by iteratively halving the step count. Nonetheless, PD’s single-step generation remains suboptimal, and the repetitive distillation procedure is time-intensive.

To address this critical issue, Song et. al. [11] proposed the consistency model (CM) for fast, single-step generation without iterative distillation. Its training goal is to reconstruct the noiseless sample in a single step from an arbitrary step on the diffusion trajectory. Our TTA framework draws inspiration from the principles underlying CM.

Besides the distinction in application domains – while CM was initially designed for image generation, we aim to enable interactive, real-time audio generation – our ConsistencyTTA introduces two innovative features requiring non-trivial technical advancements. Specifically, CM was proposed for unconditional generation; however, adapting it for conditional generation within our work demands careful consideration, primarily Classifier-Free Guidance (CFG), a subject we elaborate in Section 3. Moreover, while CM focused on pixel- [11] or spectrogram-space [28] generation, our adaptation leverages latent space for generation, thus enhancing the details of outputs without substantially increasing model size [10, 3, 1].

Shortly after this work, Luo et al. [29] used CFG-aware

latent-space CM for text-to-image and achieved exceptional quality-efficiency balance, gaining multiple implementations. This concurrent work supports our discovery and verifies our approach’s ability to make AI-assisted generation accessible.

## 2.3. Classifier-Free Guidance

CFG [12] is a highly effective method to adjust the conditioning strength for conditional generation models during inference. It significantly enhances diffusion model performance without additional training. Specifically, CFG obtains two noise estimations from the denoising network – one with conditioning (denoted as  $v_{\text{cond}}$ ) and one without (by masking the condition embedding, denoted as  $v_{\text{uncond}}$ ). The guided estimation  $v_{\text{cfg}}$  is

$$v_{\text{cfg}} = w \cdot v_{\text{cond}} + (1 - w) \cdot v_{\text{uncond}}, \quad (1)$$

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  $w > 1$ , it becomes an extrapolation.

Since CFG is external to the denoising network in diffusion models, distilling guided models is harder than unguided ones. The authors of [30] 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 (allowing selecting  $w$  during inference). Then, it performs conventional PD on this  $w$ -conditioned diffusion model. In both training stages,  $w$  is randomized. Meanwhile, our ConsistencyTTA is the first to introduce CFG into CMs.

## 3. CFG-Aware Latent-Space CM

### 3.1. Overall Setup

We select TANGO [1], a state-of-the-art (SOTA) TTA framework based on DDPM [15], as the diffusion baseline and the distillation teacher. However, we highlight that most innovations in this paper also apply to other TTA diffusion models.

Similar to TANGO, ConsistencyTTA 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 [31] that produces audio waveforms from Mel spectrograms. We only train the U-Net and freeze other components.

During training, the audio Mel spectrogram is processed by the VAE encoder, and the prompt is processed by the text encoder. The audio and text embeddings are then passed to the conditional U-Net as the input and the condition, respectively. The U-Net’s output audio embedding is used for training loss calculation. The VAE decoder and the HiFi-GAN are unused.

During inference, the audio embedding is initialized as noise, while the text encoder again produces the text embeddings. The U-Net then uses them to reconstruct a meaningful audio embedding. The VAE decoder recovers the Mel spectrogram from the generated embedding, and the HiFi-GAN produces the output waveform. The VAE encoder is unused.

### 3.2. Conditional Latent-Space Consistency Distillation

Consistency distillation (CD) aims to learn a consistency student U-Net  $f_S(\cdot)$  from the diffusion teacher module  $f_T(\cdot)$ . The inputs and outputs of  $f_S(\cdot)$  and  $f_T(\cdot)$  are latent audio embeddings. Unless mentioned otherwise,  $f_S$  and  $f_T$  have the same architecture, requiring three inputs: the noisy latent representation  $z_n$ , the time step  $n$ , and the text embedding  $e_{\text{te}}$ . Furthermore, the parameters in  $f_S$  are initialized using  $f_T$  information (more details in Section 4.3).

The student U-Net aims to generate a realistic audio embedding within a single forward pass, directly producing an estimated clean example  $\hat{z}_0$  from  $\mathbf{z}_n$ , where  $n \in \{0, \dots, N\}$  is an arbitrary step on the diffusion trajectory [11, Algorithm 2]. To achieve so, CD minimizes the training risk function

$$\mathbb{E}_{(\mathbf{z}_0, \mathbf{e}_{te}) \sim \mathcal{D}} \left[ d\left(f_S(\mathbf{z}_n, n, \mathbf{e}_{te}), f_S(\hat{\mathbf{z}}_{n-1}, n-1, \mathbf{e}_{te})\right) \right]. \quad (2)$$

Here,  $d(\cdot, \cdot)$  is a distance measure, for which we use the latent-space  $\ell_2$  distance as justified in Appendix B.3.  $\mathcal{D}$  is the data distribution, and  $U_{int}(1, N)$  denotes the discrete uniform distribution over the set  $\{1, \dots, N\}$ .  $\hat{\mathbf{z}}_{n-1}$  is the teacher diffusion model's estimation for  $\mathbf{z}_{n-1}$ . Intuitively, minimizing (2) reduces the expected distance between the student's reconstructions from two adjacent time steps on the diffusion trajectory.

The calculation for the teacher estimation  $\hat{\mathbf{z}}_{n-1}$  is solve  $\circ f_T(\mathbf{z}_n, n, \mathbf{e}_{te})$ , where solve  $\circ f_T$  is the composite function of the teacher U-Net and the ODE solver. This solver converts the U-Net's raw noise estimation to the previous time step's estimation  $\hat{\mathbf{z}}_{n-1}$ , and can be one of the samplers mentioned in Section 2.2. The authors of [11] selected the Heun solver to traverse the teacher model's diffusion trajectory during distillation. They also adopted the "Karras noise schedule", which unevenly samples time steps on the diffusion trajectory. In Section 4.2, we compare multiple solvers and noise schedules.

The literature has also considered weighting the distance  $d(\cdot, \cdot)$  in (2) based on the time step  $n$  when training diffusion models. In Appendix A.3, we analyze such weighting for CD.

### 3.3. CFG-Aware Consistency Distillation

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

**Direct Guidance** directly performs CFG on the consistency model output  $\mathbf{z}_0$  by applying (1). Since this method naïvely extrapolates/interpolates the guided and unguided  $\mathbf{z}_0$  predictions, it may move the prediction outside the manifold of realistic audio embeddings, resulting in poor generation quality.

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

$$f_T^{cfg}(\mathbf{z}_n, n, \mathbf{e}_{te}, w) = w \cdot f_T(\mathbf{z}_n, n, \emptyset) + (1 - w) \cdot f_T(\mathbf{z}_n, n, \mathbf{e}_{te}),$$

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

**Variable Guidance Distillation** mirrors fixed guidance distillation, except that the student U-Net  $f_S$  takes the CFG strength  $w$  as an additional input so that  $w$  can be adjusted *internally* during inference. To add a  $w$ -encoding condition branch to  $f_S$ , we use Fourier encoding for  $w$  following [30] and merge the  $w$  embedding into  $f_S$  similarly as the time step embedding. During distillation, 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 two-stage PD [30], with more details discussed in Appendix B.2.

### 3.4. Closed-Loop Finetuning with CLAP Score

Since ConsistencyTTA produces audio in a single neural network query, we can optimize auxiliary loss functions along with the CD objective (2). Unlike (2), the auxiliary loss can use the

generated audio waveform and can incorporate ground-truth audio and text. Hence, optimizing it provides valuable closed-loop feedback and can thus enhance the generation quality and semantics. In contrast, diffusion models cannot be trained in this closed-loop fashion. This is because their inference is iterative, and thus the generated audio is unavailable during training.

This work uses the CLAP score [13] as an example auxiliary loss function. We select it due to its consideration of ground-truth audio and text, as well as the CLAP model's high embedding quality. The CLAP score can be calculated with respect to either audio or text. We denote them as  $CLAP_A$  and  $CLAP_T$ , respectively. Specifically,  $CLAP_A$  is defined as

$$CLAP_A(\hat{\mathbf{x}}, \mathbf{x}) = \max \left\{ 100 \times \frac{\mathbf{e}_{\hat{\mathbf{x}}} \cdot \mathbf{e}_{\mathbf{x}}}{\|\mathbf{e}_{\hat{\mathbf{x}}}\| \cdot \|\mathbf{e}_{\mathbf{x}}\|}, 0 \right\}, \quad (3)$$

where  $\mathbf{e}_{\hat{\mathbf{x}}}$  and  $\mathbf{e}_{\mathbf{x}}$  are the embeddings extracted from the generated and ground-truth audio with the CLAP model.  $CLAP_T$  is defined similarly, with the CLAP text embedding used as the reference instead. During finetuning, we co-optimize three loss components: the CD objective (2),  $CLAP_A$ , and  $CLAP_T$ .

## 4. Experiments

### 4.1. Dataset, Metrics, and Model Settings

**Dataset.** For evaluation, we use AudioCaps [32], a popular and standard in-the-wild audio benchmark dataset for TTA [1, 2, 3, 8]. It is a set of human-captioned YouTube audio clips, each 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 AudioCaps without extra data, demonstrating high data efficiency. Using larger datasets may further improve our results, which we leave for future work.

**Metrics.** We use the following metrics for objective evaluation: 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 VG-Gish [33], whereas FD and KLD are the Fréchet distance and the Kullback-Leibler divergence between the PANN [34] audio embeddings.  $CLAP_A$  and  $CLAP_T$  are defined in (3).

For subjective evaluation, we collect 25 audio clips from each model, generated from the same set of prompts, and mix them with ground-truth audio samples. We instruct 20 evaluators to rate each clip from 1 to 5 in two aspects: overall audio quality ("Human Quality") and audio-text correspondence ("Human Corresp"). Further details are in Appendix B.5.

**Models.** We select FLAN-T5-Large [35] 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]. For faster training and inference, we shrink the U-Net from 866M parameters used in [1] to 557M. As shown in Table 1, this smaller TANGO model performs similarly to the checkpoint from [1]. ConsistencyTTA is subsequently distilled from this smaller model. Additional details about our model, training, and evaluation setups are in Appendices B.3, B.4 and B.5 respectively. In all tables, "CFG  $w$ " is the CFG weight and "# Queries" indicates the number of inference U-Net queries.

### 4.2. Main Evaluation Results

Table 1 presents our main results, which compares ConsistencyTTA with or without CLAP-finetuning against several SOTA diffusion baseline models, namely AudioLDM [3] and TANGO

Table 1: **Main results:** ConsistencyTTA achieves a 400x computation reduction while achieving similar objective and subjective audio quality as SOTA diffusion methods. **Bold numbers** indicate the best ConsistencyTTA results.

		U-Net # Params	CLAP Finetuning	CFG $w$	# Queries ( $\downarrow$ )	Human Quality ( $\uparrow$ )	Human Corresp ( $\uparrow$ )	CLAP <sub>T</sub> ( $\uparrow$ )	CLAP <sub>A</sub> ( $\uparrow$ )	FAD ( $\downarrow$ )	FD ( $\downarrow$ )	KLD ( $\downarrow$ )
Diffusion Baselines	AudioLDM-L	739M	<b>X</b>	2	400	-	-	-	-	2.08	27.12	1.86
	TANGO	866M	<b>X</b>	3		-	-	24.10	72.85	1.631	20.11	1.362
	Teacher	557M	<b>X</b>	3	400	4.136	4.064	24.57	72.79	1.908	19.57	1.350
ConsistencyTTA (ours)		559M	<b>X</b> <b>✓</b>	5 4	<b>1</b>	<b>3.902</b> 3.830	4.010 <b>4.064</b>	22.50 <b>24.69</b>	72.30 <b>72.54</b>	2.575 <b>2.406</b>	22.08 <b>20.97</b>	<b>1.354</b>
Ground-Truth		-	-	-	-	4.424	4.352	26.71	100.0	0.000	0.000	0.000

Diffusion Baselines Details: AudioLDM-L: numbers reported in [3]. TANGO: checkpoint from [1], tested by us. Teacher: A smaller TANGO model trained by us, used as ConsistencyTTA’s distillation teacher.

Table 2: *Ablation study on guidance weights, distillation techniques, solvers, noise schedules, training lengths, and initializations.*

Guidance Method	Solver	Noise Schedule	CFG $w$	Initialization	# Queries ( $\downarrow$ )	FAD ( $\downarrow$ )	FD ( $\downarrow$ )	KLD ( $\downarrow$ )
Unguided	DDIM	Uniform	1	Unguided	1	13.48	45.75	2.409
Direct Guidance	DDIM	Uniform	3	Unguided	2	8.565	38.67	2.015
	Heun	Karras				7.421	39.36	1.976
Fixed Guidance Distillation	Heun	Karras	3	Unguided	1	5.702	33.18	1.494
		Uniform		Unguided		4.168	28.54	1.384
		Uniform		Guided		3.859	<b>27.79</b>	1.421
Variable Guidance Distillation	Heun	Uniform	4	Guided	1	3.180	27.92	1.394
			6			<b>2.975</b>	28.63	<b>1.378</b>

[1]. Distillation runs are 60 epochs, CLAP-finetuning uses 10 additional epochs, and inference uses BF16 precision.

Table 1 shows that ConsistencyTTA’s generated audio quality is similar to that of SOTA diffusion models in all objective and subjective metrics. Notably, ConsistencyTTAs’ FD and KLD even surpass the reported numbers from both AudioLDM and TANGO (which reported 24.53 FD and 1.37 KLD). We encourage readers to listen to the generations on our website<sup>1</sup>.

All diffusion baseline models use 200 inference steps following [3, 1], each step needing two noise estimations due to CFG, summing to 400 network queries per generation. Hence, we can conclude that with minimal performance drop, ConsistencyTTA reduces the U-Net queries by a factor of 400.

Table 1 also shows that closed-loop-finetuning ConsistencyTTA by optimizing the CLAP scores improves not only the CLAP scores but also FAD and FD. This cross-metric agreement implies that the observed improvement is due to all-around generation quality enhancement, not overfitting the optimized metric. With CLAP-finetuning, the text-audio correspondence also sees an improvement, with the subjective Human Corresp score reaching the same level as the teacher diffusion model and the objective CLAP<sub>T</sub> even exceeding that of the teacher. This observation supports our hypothesis that adding the prompt-aware CLAP<sub>T</sub> to the optimization objective provides closed-loop feedback to help align generated audio with the prompt.

In Appendix A.1, we show that ConsistencyTTA generates better audio faster than existing training-free diffusion acceleration methods. In Appendix A.2, we discuss the significant 72x real-world computing time reduction of ConsistencyTTA.

### 4.3. Ablation Study

Table 2 evaluates ConsistencyTTA across different distillation settings. “Guided initialization” initializes ConsistencyTTA weights with a CFG-aware diffusion model (similar to [30]), whereas “unguided initialization” uses the original TANGO teacher weights. All U-Nets have 557M parameters, except the variable guidance one which uses 2M extra for  $w$ -encoding. Distillation spans 40 epochs and inference uses FP32 precision.

Table 2 shows that distilling with fixed or variable guidance

significantly improves all metrics over direct or no guidance, highlighting the importance of CFG-aware distillation.

While a CFG weight of 3 is ideal for the teacher diffusion model, the optimal  $w$  is larger for the variable guidance distilled model, aligning with the observations in [30]. In Appendix A.4, we confirm this observation by analyzing how the generation quality of the ConsistencyTTA models in Table 1 varies with  $w$ .

Meanwhile, using the more accurate Heun solver to traverse the teacher model’s diffusion trajectory during distillation outperforms distilling with the simpler DDIM solver. In contrast to [11], the uniform noise schedule is preferred over the Karras schedule, with the former achieving superior FAD, FD, and KLD (detailed discussions in Appendix B.1). Finally, guided initialization improves FD and FAD but slightly sacrifices KLD.

### 4.4. Audio Generation Diversity

ConsistencyTTA produces diverse generations as do diffusion models. Different random seeds (different initial Gaussian embeddings) produce noticeably different audio. To demonstrate, we present the generated waveforms from the first 50 AudioCaps test prompts with four different seeds at<sup>2</sup>. We display the corresponding spectrograms, along with quantitative generation diversity analyses, in Appendix A.5.

## 5. Conclusion

This work proposes ConsistencyTTA, a innovative approach leveraging consistency models to accelerate diffusion-based TTA models hundreds of times while maintaining generation quality and diversity. Central to this vast acceleration are two innovations: *CFG-aware latent CM* and *closed-loop CLAP-finetuning*. The former introduces CFG into the training process, significantly enhancing the performance of conditional CMs. The latter utilizes the differentiability of ConsistencyTTA to provide crucial text-aware closed-loop feedback to the underlying network. As a result, ConsistencyTTA enables TTA generation in realtime settings, and significantly broadens its accessibility for AI researchers, audio professionals, and enthusiasts.

<sup>2</sup>[consistency-tta.github.io/diversity-anony](https://consistency-tta.github.io/diversity-anony)

## 6. References

- [1] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, “Text-to-audio generation using instruction-tuned LLM and latent diffusion model,” *arXiv preprint arXiv:2304.13731*, 2023.
- [2] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, “Diffsound: Discrete diffusion model for text-to-sound generation,” *Transactions on Audio, Speech, and Language Processing*, 2023.
- [3] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “AudioLDM: Text-to-audio generation with latent diffusion models,” *arXiv preprint arXiv:2301.12503*, 2023.
- [4] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, “AudioLDM 2: Learning holistic audio generation with self-supervised pretraining,” *arXiv preprint arXiv:2308.05734*, 2023.
- [5] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, “Make-an-Audio: Text-to-audio generation with prompt-enhanced diffusion models,” *arXiv preprint arXiv:2301.12661*, 2023.
- [6] J. Huang, Y. Ren, R. Huang, D. Yang, Z. Ye, C. Zhang, J. Liu, X. Yin, Z. Ma, and Z. Zhao, “Make-an-Audio 2: Temporal-enhanced text-to-audio generation,” *arXiv preprint arXiv:2305.18474*, 2023.
- [7] Z. Tang, Z. Yang, C. Zhu, M. Zeng, and M. Bansal, “Any-to-any generation via composable diffusion,” *arXiv preprint arXiv:2305.11846*, 2023.
- [8] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “AudioGen: Textually guided audio generation,” in *International Conference on Learning Representations*, 2023.
- [9] S. Forsgren and H. Martiros, “Riffusion - stable diffusion for real-time music generation,” URL <https://riffusion.com>, 2022.
- [10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Conference on Computer Vision and Pattern Recognition*, 2022.
- [11] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” in *International Conference on Machine Learning*, 2023.
- [12] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.
- [13] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, “CLAP: learning audio concepts from natural language supervision,” in *International Conference on Acoustics, Speech and Signal Processing*, 2023.
- [14] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *International Conference on Machine Learning*, 2015.
- [15] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems*, 2020.
- [16] T. Karras, M. Aittala, T. Aila, and S. Laine, “Elucidating the design space of diffusion-based generative models,” in *Advances in Neural Information Processing Systems*, 2022.
- [17] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, “Noise2Music: Text-conditioned music generation with diffusion models,” *arXiv preprint arXiv:2302.03917*, 2023.
- [18] Y. Bai, U. Garg, A. Shanker, H. Zhang, S. Parajuli, E. Bas, I. Filipovic, A. N. Chu, E. D. Fomitcheva, E. Branson *et al.*, “Let’s go shopping (LGS)–web-scale image-text dataset for visual concept understanding,” *arXiv preprint arXiv:2401.04575*, 2024.
- [19] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention*, 2015.
- [20] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” *arXiv preprint arXiv:2010.02502*, 2020.
- [21] L. Euler, *Institutionum calculi integralis.* impensis Academiae imperialis scientiarum, 1824, vol. 1.
- [22] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. 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] ———, “DPM-solver++: Fast solver for guided sampling of diffusion probabilistic models,” *arXiv preprint arXiv:2211.01095*, 2022.
- [24] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, “Pseudo numerical methods for diffusion models on manifolds,” in *International Conference on Learning Representations*, 2022.
- [25] F. Bao, C. Li, J. Zhu, and B. Zhang, “Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models,” in *International Conference on Learning Representations*, 2021.
- [26] T. Salimans and J. Ho, “Progressive distillation for fast sampling of diffusion models,” in *International Conference on Learning Representations*, 2021.
- [27] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach, “Adversarial diffusion distillation,” *arXiv preprint arXiv:2311.17042*, 2023.
- [28] Z. Ye, W. Xue, X. Tan, J. Chen, Q. Liu, and Y. Guo, “CoMo-Speech: One-step speech and singing voice synthesis via consistency model,” *arXiv preprint arXiv:2305.06908*, 2023.
- [29] S. Luo, Y. Tan, L. Huang, J. Li, and H. Zhao, “Latent consistency models: Synthesizing high-resolution images with few-step inference,” *arXiv preprint arXiv:2310.04378*, 2023.
- [30] C. Meng, R. Rombach, R. Gao, D. Kingma, S. Ermon, J. Ho, and T. Salimans, “On distillation of guided diffusion models,” in *Conference on Computer Vision and Pattern Recognition*, 2023.
- [31] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *Advances in Neural Information Processing Systems*, 2020.
- [32] C. D. Kim, B. Kim, H. Lee, and G. Kim, “AudioCaps: Generating captions for audios in the wild,” in *Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.
- [33] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, “CNN architectures for large-scale audio classification,” in *International Conference on Acoustics, Speech and Signal Processing*, 2017.
- [34] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. 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.
- [35] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, “Scaling instruction-finetuned language models,” *arXiv preprint arXiv:2210.11416*, 2022.
- [36] R. Huang, M. W. Lam, J. Wang, D. Su, D. Yu, Y. Ren, and Z. Zhao, “Fastdiff: A fast conditional diffusion model for high-quality speech synthesis,” in *International Joint Conference on Artificial Intelligence*, 2022.
- [37] T. Hang, S. Gu, C. Li, J. Bao, D. Chen, H. Hu, X. Geng, and B. Guo, “Efficient diffusion training via min-snR weighting strategy,” *arXiv preprint arXiv:2303.09556*, 2023.
- [38] B. McFee, “ResamPy: efficient sample rate conversion in python,” *Journal of Open Source Software*, vol. 1, no. 8, p. 125, 2016.
- [39] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsich, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, “TorchAudio: Building blocks for audio and speech processing,” *arXiv preprint arXiv:2110.15018*, 2021.
- [40] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. 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.
- [41] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio Set: An ontology and human-labeled dataset for audio events,” in *International Conference on Acoustics, Speech and Signal Processing*, 2017.

Table 3: Compare our ConsistencyTTA model with training-free diffusion acceleration methods, specifically improved ODE solvers. All diffusion models use the same TANGO weights as in Table 1 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 [16] with the Heun solver.

Model Type	Solver	# Queries ( $\downarrow$ )	FAD ( $\downarrow$ )	FD ( $\downarrow$ )	KLD ( $\downarrow$ )
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 Experiments

### A.1. Comparison with Training-Free Acceleration Methods

This section compares consistency models with diffusion acceleration methods that do not require tuning 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 3, we compare our single-step consistency models with training-free methods.

As shown in Table 3, 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 reasonable audio. Among these solvers, Heun achieves the best generation quality, but is still worse than the single-step ConsistencyTTA. 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 to those presented in Table 3, 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 3. 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. Real-World 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.

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, our results are timed with off-the-shelf PyTorch code. Real-world speed-up can be even more prominent with implementation optimizations, approaching the hundreds-fold theoretical acceleration.

### A.3. 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 [37] is one of the latest strategies. The Min-SNR calculation depends on whether the diffusion model predicts the clean example  $z_0$ , the additive noise  $\epsilon$ , or the noise velocity  $v$ .

This work investigates how Min-SNR affects CD. Since consistency models predict the clean sample  $z_0$ , we use the Min-SNR formulation for  $z_0$ -predicting diffusion models, which is  $\omega(n) = \min\{\text{SNR}(t_n), \gamma\}$ , where  $\omega(n)$  is the loss weight for the  $n^{\text{th}}$  time step,  $\text{SNR}(t)$  is the SNR at time  $t$ ,  $t_n$  is the time corresponding to the  $n^{\text{th}}$  time step, and  $\gamma$  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  $t$ .

We analyze the effect of Min-SNR with the following setting: fixed guidance distillation with  $w = 3$ , Heun solver for the teacher model with Uniform noise schedule, and Unguided initialization. Without Min-SNR, the FAD, FD, and KLD are 4.168, 28.54, and 1.384. With Min-SNR, they are 3.766, 27.74, and 1.443 (lower is better).

We can therefore conclude that Min-SNR loss weighting improves FD and FAD but slightly sacrifices KLD. Hence, we apply Min-SNR to the models in our main results (Table 1).

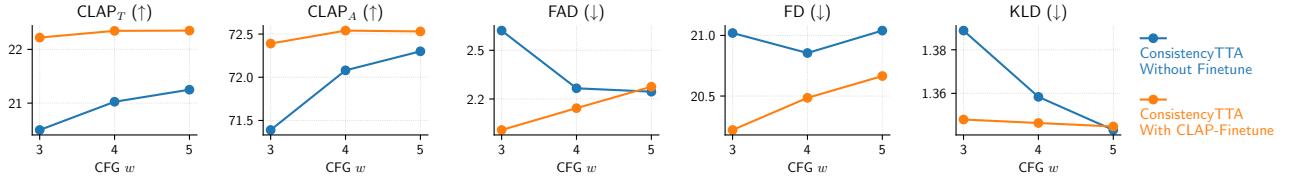
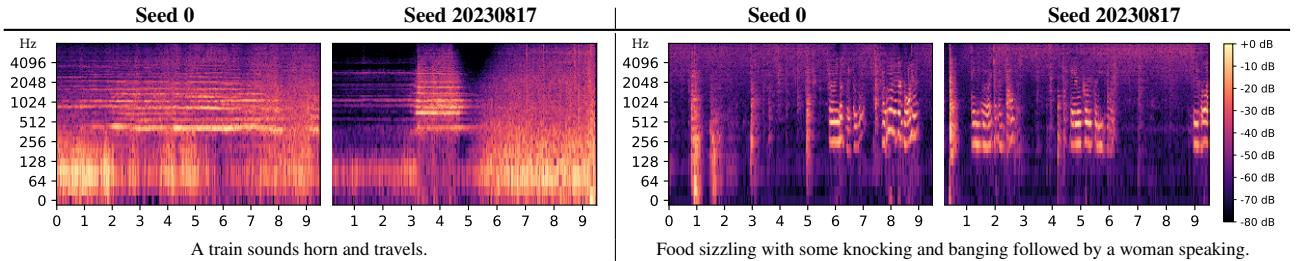


Figure 2: ConsistencyTTA checkpoints in Table 1 with different CFG weights.

Table 4: The generated audio noticeably varies with different random seeds. The horizontal axis is time in seconds.



#### A.4. Ablation on the CFG Weight $w$

In this section, we investigate how the CFG weight  $w$  affects the ConsistencyTTA models presented in Table 1. Intuitively, a larger  $w$  value indicates a stronger text conditioning. Recall that with ConsistencyTTA,  $w$  is an input to the latent-space consistency generation U-Net as a result of the variable-guidance distillation process. Here, we consider three values for  $w$ : 3, 4, and 5, and present the results in Figure 2. We can observe the following:

- For all five objective metrics, ConsistencyTTA after CLAP-finetuning outperforms the model without finetuning for almost all values of  $w$ .
- CLAP<sub>A</sub>, CLAP<sub>T</sub>, and KLD improve as  $w$  increase from 3 to 5 for both checkpoints. The CLAP score improvement especially makes sense because a stronger text condition should improve the generations semantically, enhancing the correspondence with the text and ground-truth audio.
- When  $w$  increases, the FAD improves for the model without finetuning but worsens for the model after CLAP-finetuning.
- For the model without finetuning,  $w = 4$  achieves the best FD. For the CLAP-finetuned model, FD worsens as  $w$  increases.

Based on these observations, we can summarize two main conclusions. First, ConsistencyTTA generally prefers a larger  $w$  value than its diffusion teacher model, for which the optimal  $w$  is 3. This makes sense because for the diffusion model, CFG is an extrapolation outside the neural network, and hence using a large  $w$  faces the risk of moving outside the manifold of realistic audio embeddings. Meanwhile, CFG is integral to ConsistencyTTA and does not have this problem. A larger  $w$  value can thus be used to improve the semantic understanding. Among the two ConsistencyTTA models, the one without finetuning prefers even larger  $w$  values than the CLAP-finetuned one. Second, when  $w$  is between 3 and 5, adjusting  $w$  largely results in a CLAP<sub>A</sub>/CLAP<sub>T</sub>/KLD versus FD/FAD trade-off. Selecting  $w = 5$  for the non-finetuned model and  $w = 4$  for the finetuned model results in a balance across all metrics.

#### A.5. More Generation Diversity Evidences

The generation diversity of ConsistencyTTA 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 [11]). Hence, consistency models’ generation diversity is on par with diffusion models’, which is known to be highly diverse.

This section presents the generated spectrograms from the consistency models using different seeds, demonstrating that ConsistencyTTA simultaneously achieves efficient generation and diversity, a goal previous models struggled to reach. Table 4 presents the generated spectrograms (calculated via performing STFT on the generated waveforms) from two example prompts with two different seeds, whereas Figure 3 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-anony](https://consistency-tta.github.io/diversity-anony)). 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. ConsistencyTTA reaches an IS of 8.29/8.88 before/after CLAP finetuning, surpassing AudioLDM [3], which reported 8.13, and TANGO [1], which achieved 8.26 (test by us since [1] did not report IS).

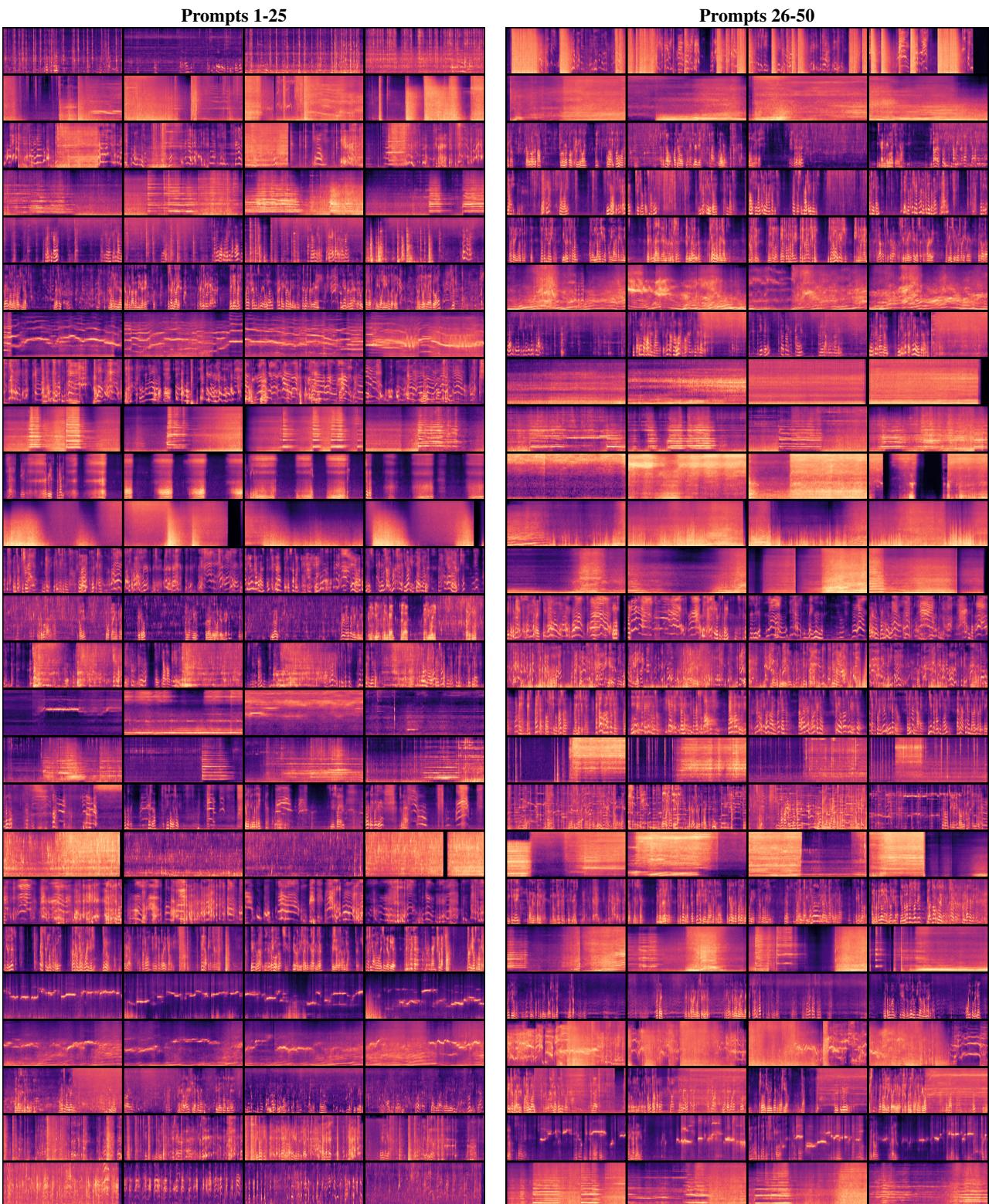


Figure 3: Consistency model generated Mel spectrograms from the first 50 AudioCaps prompts with four different seeds. Each row corresponds to a prompt, and each column corresponds to a seed. The generations from a prompt with different seeds are correlated but distinctly different.

## B. Additional Discussions and Details

### B.1. Additional Discussions Regarding the Teacher Solver

Table 2 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 [11]. Moreover, as shown in Table 3, 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.

Table 2 additionally suggests that distilling with the uniform noise schedule outperforms the Karras schedule (DDIM+uniform  $\approx$  Heun+Karras < Heun+uniform). This observation is surprising because previous work [11] suggested using the Karras schedule. Our explanation for this difference is that TANGO was trained with the uniform schedule, whereas the teacher models in [11] were trained with the Karras schedule. It is likely beneficial to use the same noise schedule during distillation and diffusion teacher training.

### B.2. Relationship to Two-Stage Progressive Distillation

Unlike PD in [30], 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 [30] 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.

### B.3. Model Details

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 finetuning. The “CD target network” (see [11] 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.

### B.4. Training Details

The ConsistencyTTA models in the main results (Table 1) use the best setting concluded from our ablation study: variable guidance distillation, Heun teacher solver, uniform noise schedule, guided initialization, and Min-SNR loss weighting. All runs use  $N = 18$  diffusion discretization steps during distillation as in [11].

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 [38] for all resampling procedures unless the resampling step needs to be differentiable. Specifically, CLAP finetuning requires differentiable resampling, and we use TorchAudio [39] instead.

Regarding the distance measure  $d(\cdot, \cdot)$  introduced in (2), the authors of [11] considered several options 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 feature distance at several layers of this deep model) as the optimization objective produced higher generation quality than using the pixel-level  $\ell_2$  or  $\ell_1$  distance. However, since our latent diffusion model already operates in a latent feature space, using the  $\ell_2$  distance in this latent space is the most logical option.

### B.5. 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 1 and 2, we truncate the generated audio to 9.70 seconds. Table 2 uses the full 10.24 seconds. The ground-truth reference waveforms are kept as-is. For CLAP<sub>A</sub> and CLAP<sub>T</sub> calculations, we use the CLAP checkpoint from [40] trained on LAION-Audio-630k [40], AudioSet [41], and music.

The human evaluation results in Table 1 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 to 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-anony](https://consistency-tta.github.io/evaluation-anony).