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

Anonymous submission to Interspeech 2024

Abstract

Diffusion models power a vast majority of text-to-audio (TTA) generation methods. Unfortunately, they suffer from unacceptably slow inference, requiring hundreds of queries to the underlying denoising network per generation. This work proposes ConsistencyTTA, a novel TTA framework that generates within a single non-autoregressive neural network query. To achieve this, we propose "CFG-aware latent consistency model", which moves consistency generation into a latent space and incorporates classifier-free guidance (CFG) into model training. Unlike diffusion models, ConsistencyTTA's single-step generation makes its generated audio available during training. We leverage this advantage to finetune ConsistencyTTA end-to-end with audio-space text-aware metrics, such as the CLAP score, further enhancing the generations. Our objective and subjective evaluation with the AudioCaps dataset shows that ConsistencyTTA retains diffusion models' high generation quality and diversity while reducing the inference computation 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, recently gained significant popularity [1, 2, 3, 4, 5, 6, 7, 8, 9]. Many TTA models are based on latent diffusion models (LDM) [10], which are famous for superior generation quality and diversity [10]. Unfortunately, LDMs suffer from slow inference as they require iterative neural network queries, posing latency and computation challenges. Hence, accelerating diffusion-based TTA will make such technologies vastly more accessible and reduce their carbon footprint, facilitating AI-assisted real-world media creation.

This work proposes *ConsistencyTTA*, which accelerates diffusion-based TTA hundreds of times with a novel *CFG-aware latent-space consistency model* that only requires a single non-autoregressive neural network query per generation. ConsistencyTTA moves consistency model [11] generation into a latent space and incorporates classifier-free guidance (CFG) [12] into the training process to significantly enhance conditional generation quality. We analyze three approaches for incorporating CFG: direct guidance, fixed guidance, and variable guidance. To our knowledge, we are the first to introduce CFG into CMs, not only for TTA but also for general content generation.

A distinct advantage of consistency models (CM) is that the generated audio is available during training. In contrast, diffusion models' generations are usually unavailable during training due to their recurrent inference process. To this end, we finetune ConsistencyTTA in an end-to-end fashion with audio quality and audio-text correspondence objective functions. We use the CLAP score as an example objective and verify the improved generation quality and text correspondence.

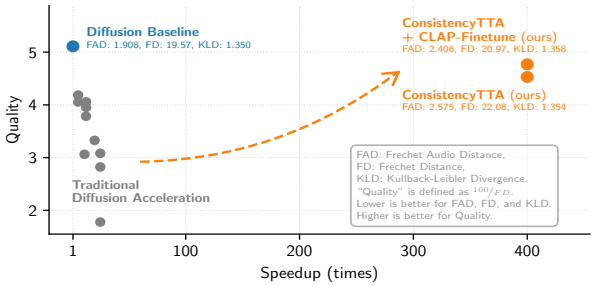


Figure 1: *ConsistencyTTA achieves a 400x computation reduction compared with a diffusion baseline model while sacrificing much less quality than traditional acceleration methods.*

Our extensive experiments, summarized by Figure 1, show that ConsistencyTTA simultaneously achieves quality, speed, and diversity. 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. We encourage the reader to listen to our generated examples at `consistency-tta.github.io/demo-anony`.

When implemented with standard Python libraries, ConsistencyTTA takes an average of 9.1 seconds to generate one minute of audio *on a laptop computer*. In contrast, a representative diffusion method [1] needs more than a minute on a state-of-the-art A100 GPU to do the same (details in Appendix B.5).

Shortly after this work, Luo et al. [13] 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. Background and Related Work

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

2.1. Diffusion Models

Diffusion models [14, 15], known for their diverse, high-quality generations, have rapidly gained popularity among conditional and unconditional vision and audio generation tasks

Additional results, details, and discussions presented in the supplemental material `consistency-tta.github.io/report.pdf`.

[10, 16, 3, 17, 18]. In the vision domain, while pixel-level diffusion (e.g., EDM [16]) performs well on small image sizes, producing larger images usually requires LDMs [10], where the diffusion process occurs in a latent space. In the audio domain, generation can be categorized into speech, music, and in-the-wild audio creation. This paper considers the in-the-wild setting, where the goal is to produce 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 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 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 (training-free) and distillation methods (training-based).

Improved samplers reduce the number of inference steps N of already-trained 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 from the hundreds required by vanilla DDPM [15] to 10-50. However, they struggle to reduce N below 10.

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 below 10 [26, 11, 27]. One example 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 [11] has been proposed for single-step fast generation without iterative distillation. The training goal of CMs is to reconstruct the noiseless image within a single step from an arbitrary step on the diffusion trajectory.

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

Previously, CMs focused on pixel [11] or spectrogram-space [28] generation. Meanwhile, diffusion models demonstrated that latent-space generation produces much finer details without excessively increasing model size [10, 3, 1]. This work moves CM generation into a latent space and demonstrates its effectiveness in text-conditioned in-the-wild audio generation.

CMs were originally proposed for unconditional generation [11]. Conditional generation in this work demands additional considerations, mainly CFG, which we discuss in Section 3.

2.3. Classifier-Free Guidance

CFG [12] 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

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

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 $w > 1$, CFG becomes extrapolation.

Since CFG is external to the denoising network in diffusion models, distillating guided models is more complex than unguided ones. The authors of [29] 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 for selecting w during inference). Then, it performs PD on the 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, as the diffusion baseline and the distillation teacher. However, we highlight that most innovations in this paper also apply to other diffusion-based TTA 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 [30] 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 to produce a latent representation, and the prompt is transformed by the text encoder into an 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 encoder again produces text embeddings, guiding the U-Net to reconstruct a latent audio representation. The VAE decoder then recovers the Mel spectrogram from the generated embedding, and the HiFi-GAN vocoder produces the output waveform. The VAE encoder is unused.

3.2. Conditional Latent-Space Consistency Distillation

The goal of consistency distillation (CD) is to learn a consistency student U-Net $f_{\text{S}}(\cdot)$ from the diffusion teacher module $f_{\text{T}}(\cdot)$ in TANGO. Unlike pixel-space or spectrogram-space CMs [11, 28], the inputs and outputs of $f_{\text{S}}(\cdot)$ and $f_{\text{T}}(\cdot)$ are latent audio embeddings. 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 audio embedding within a single forward pass, directly producing an estimated clean example $\hat{\boldsymbol{z}}_0$ from $\boldsymbol{z}_n$, where $n \in \{0, \ldots, N\}$ is an arbitrary step on the diffusion trajectory [11, Algorithm 2]. To achieve so, CD minimizes the function

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

where $d(\cdot,\cdot)$ is a distance measurement, $\mathcal{D}$ is the training dataset, $\mathrm{U}_{\mathrm{int}}(1,N)$ denotes the discrete uniform distribution supported over the set $\{1,\dots,N\}$, and $\hat{\boldsymbol{z}}_{n-1} = \mathrm{solve}\circ f_{\mathrm{T}}(\boldsymbol{z}_n,n,\boldsymbol{e}_{\mathrm{te}})$ is the teacher diffusion model's estimation for $\boldsymbol{z}_{n-1}$. Here, $\mathrm{solve}\circ f_{\mathrm{T}}$ denotes the composite function of the teacher denoising U-Net and the solver that converts this U-Net raw output to the previous time step's estimation $\hat{\boldsymbol{z}}_{n-1}$. We use the latent-space ℓ_2 distance as $d(\cdot,\cdot)$, as justified in Appendix B.3. Intuitively, the risk function (2) measures the expected distance between the student's reconstructions from two adjacent time steps on the diffusion trajectory.

The authors of [11] used the Heun solver to traverse the teacher model's diffusion trajectory during distillation and adopted "Karras noise schedule", which unevenly samples time steps on the diffusion trajectory. In Section 4.2, we empirically investigate 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 $\boldsymbol{z}_0$ 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. The training risk function is still (2), but $\hat{\boldsymbol{z}}_{n-1}$ is replaced with the estimation after CFG. Specifically, $\hat{\boldsymbol{z}}_{n-1}$ becomes $\mathrm{solve}\circ f_{\mathrm{T}}^{\mathrm{cfg}}(\boldsymbol{z}_n,n,\boldsymbol{e}_{\mathrm{te}},w)$, where the guided teacher output $f_{\mathrm{T}}^{\mathrm{cfg}}$ is

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

with $\varnothing$ denoting the masked language token. Here, w is fixed to the value that optimizes teacher generation (3 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 [29] 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 [29], with the details in Appendix B.2.

3.4. End-to-End Finetuning with CLAP

Since ConsistencyTTA 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 [31] as an example of finetuning loss functions. It is defined as

$$\mathrm{ClapScore}(\hat{\boldsymbol{x}},\boldsymbol{x}) = \max\Big\{100\times\frac{\mathbf{e}_{\hat{\boldsymbol{x}}}\cdot\mathbf{e}_{\boldsymbol{x}}}{\|\mathbf{e}_{\hat{\boldsymbol{x}}}\|\cdot\|\mathbf{e}_{\boldsymbol{x}}\|},0\Big\}, \quad (3)$$

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 the CLAP model's superior embedding quality arising from its 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, co-optimizing the CLAP score provides valuable closed-loop feedback to ConsistencyTTA.

4. Experiments

4.1. Dataset, Metrics, 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, without extra training data, demonstrating high data efficiency. Using larger datasets may further improve our results, which we leave for future work.

For objective evaluation, we use the following metrics: FAD, FD, KLD, $\mathrm{CLAP_A}$, and $\mathrm{CLAP_T}$. The former four use the ground-truth audio as the reference, whereas $\mathrm{CLAP_T}$ uses the text. Specifically, FAD is the Fréchet distance between generated and ground-truth audio embeddings extracted by VGGish [33], whereas FD and KLD are the Fréchet distance and the Kullback-Leibler divergence between the PANN [34] audio embeddings. $\mathrm{CLAP_A}$ and $\mathrm{CLAP_T}$ are the CLAP scores with respect to the ground-truth audio and the textual prompt.

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.

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. For end-to-end CLAP finetuning, we co-optimize three loss components: the consistency loss (2), $\mathrm{CLAP_A}$, and $\mathrm{CLAP_T}$. Additional details about our model, training, and evaluation setups are in Appendices B.3, B.4 and B.5.

In all tables, "CFG w" is the CFG weight and "# Queries" indicates the number of inference U-Net queries.

4.2. Main Evaluation Results

Our main results are presented in Table 1, which compares ConsistencyTTA with or without CLAP-finetuning against several SOTA diffusion baseline models, namely AudioLDM [3] and TANGO [1]. Distillation runs are 60 epochs, CLAP-finetuning uses 10 additional epochs, and inference uses BF16 precision.

Table 1 demonstrates that ConsistencyTTA's generated audio quality is similar to that of diffusion models in terms of all objective and subjective metrics. Notably, the FD

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 (↓)	Human Quality (↑)	Human Corresp (↑)	CLAP_T (↑)	CLAP_A (↑)	FAD (↓)	FD (↓)	KLD (↓)
Diffusion Baselines	AudioLDM-L	739M	✗	2	400	-	-	-	-	2.08	27.12	1.86
	TANGO	866M	✗	3		-	-	24.10	72.85	1.631	20.11	1.362
	Teacher	557M	✗	3	400	4.136	4.064	24.57	72.79	1.908	19.57	1.350
ConsistencyTTA (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

Diffusion Baselines Details: AudioLDM-L: numbers reported in [3]. TANGO: checkpoint from [1], tested by us. Teacher: A 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 (↓)	FAD (↓)	FD (↓)	KLD (↓)
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	**27.79**	1.421
Variable Guidance Distillation	Heun	Uniform	4	Guided	1	3.180	27.92	1.394
			6			**2.975**	28.63	**1.378**

and KLD even surpass the reported numbers of both AudioLDM and TANGO (which reported 24.53 FD and 1.37 KLD). We encourage the reader to listen to the generations at `consistency-tta.github.io/demo-anony`.

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 end-to-end-finetuning ConsistencyTTA by optimizing the CLAP scores improves not only the CLAP scores but also FAD and FD. This multi-metric improvement implies an all-around generation quality enhancement and an absence of overfitting to the optimized metric. With CLAP-finetuning, the text-audio correspondence also sees an improvement, with the subjective Human Corresp reaching the same level as the teacher diffusion model and the objective CLAP_T even exceeding that of the teacher. This observation supports our hypothesis that adding the prompt-aware CLAP_T 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

In Table 2, we compare the performance of ConsistencyTTA under various distillation settings. "Guided initialization" refers to initializing ConsistencyTTA weights with a CFG-aware diffusion model (similar to [29]), whereas "unguided initialization" initializes with the unmodified TANGO teacher weights. All U-Nets have 557M parameters, except the variable guidance one which uses 2M extra for w-encoding. Distillation runs are 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 the CFG weight $w = 3$ is ideal for the teacher diffusion model, the optimal w becomes larger for the variable guidance distilled model, aligning with the observations in [29]. In Figure 2 in Appendix A.4, we confirm this phenomenon by demonstrating how the generation quality of the ConsistencyTTA checkpoints in Table 1 changes with w.

Meanwhile, using the more accurate Heun solver to traverse the teacher model's diffusion trajectory for distillation outperforms the simpler DDIM solver. Surprisingly, the uniform noise schedule is preferred over the Karras schedule, as the former achieves superior FAD, FD, and KLD (see Appendix B.1 for more detailed discussions). 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 seeds at `consistency-tta.github.io/diversity-anony`, and display the corresponding spectrograms in Appendix A.5, where we also provide quantitative evidence.

5. Conclusion

This work proposes ConsistencyTTA, a novel approach to accelerate the core module of diffusion-based TTA models hundreds of times based on consistency models with negligible generation quality and diversity degradation. At the core of this vast speed-up are two innovations: *CFG-aware latent CM* and *end-to-end CLAP-finetuning*. The former introduces conditional CFG into the training process, vastly promoting the performance of CMs. The latter leverages the unique differentiability of ConsistencyTTA to provide important text-aware closed-loop feedback to the underlying neural network. With these novel training configurations, ConsistencyTTA makes AI-assisted audio generation more efficient and accessible than ever, for AI researchers, audio professionals, and hobbyists alike.

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] 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.

[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, "CoMoSpeech: One-step speech and singing voice synthesis via consistency model," *arXiv preprint arXiv:2305.06908*, 2023.

[29] 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.

[30] 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.

[31] 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.

[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. Puhrsch, 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 (↓)	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 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 $\boldsymbol{z}_0$, 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 $\omega(n) = \min\{\mathrm{SNR}(t_n), \gamma\}$, where $\omega(n)$ is the loss weight for the n^{th} time step, $\mathrm{SNR}(t)$ 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, $\mathrm{SNR}(t)$ 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).

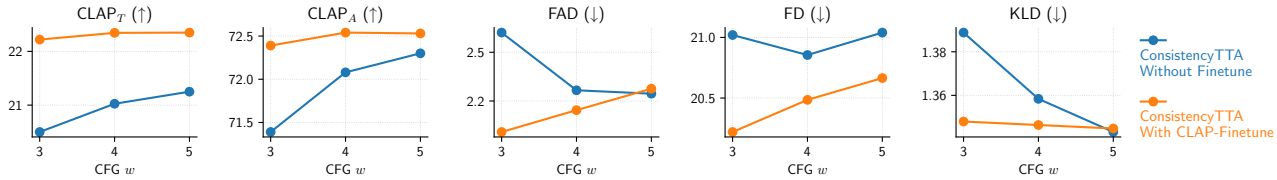
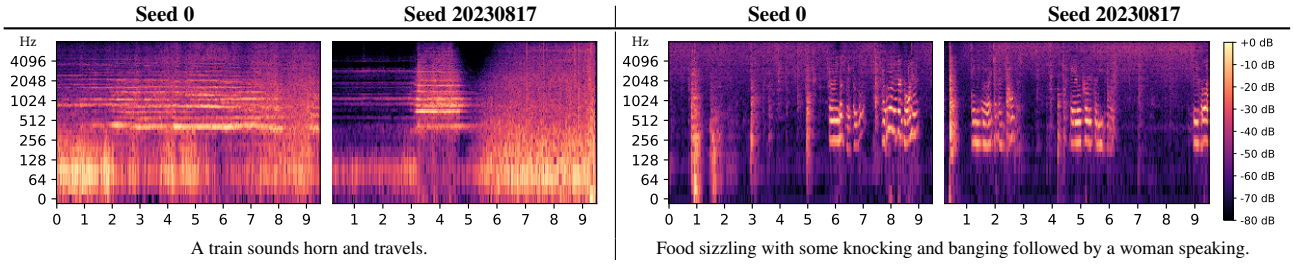


Figure 2: *ConsistencyTTA checkpoints in Table 1 with different CFG weights.*

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$_\text{A}$, CLAP$_\text{T}$, 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$_\text{A}$/CLAP$_\text{T}$/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`). 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).

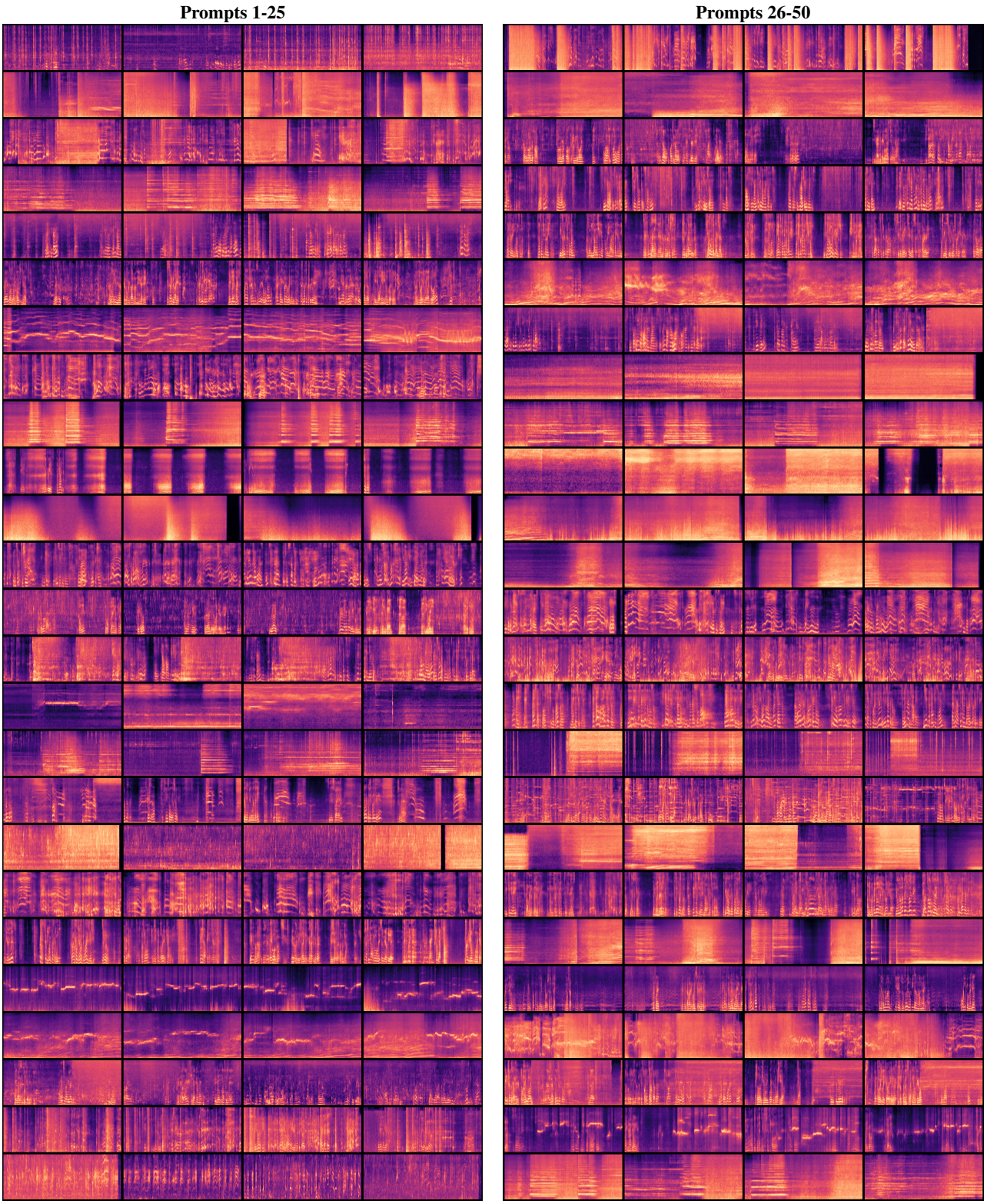


Figure 3: *Consistency model generated Mel spectrograms from the first 50 AudioCaps prompts with four different seeds. Each row corresponds to a prompt, and each column corresponds to a seed. The generations from a prompt with different seeds are correlated but distinctly different.*

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 [29], 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 [29] 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 ℓ_2 or ℓ_1 distance. However, since our latent diffusion model already operates in a latent feature space, using the ℓ_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_A and CLAP_T 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`.