Diffusion-based Vocoding for Real-Time Text-To-Speech

Lukas Gardberg

LTH

March 23, 2023

Presentation Layout

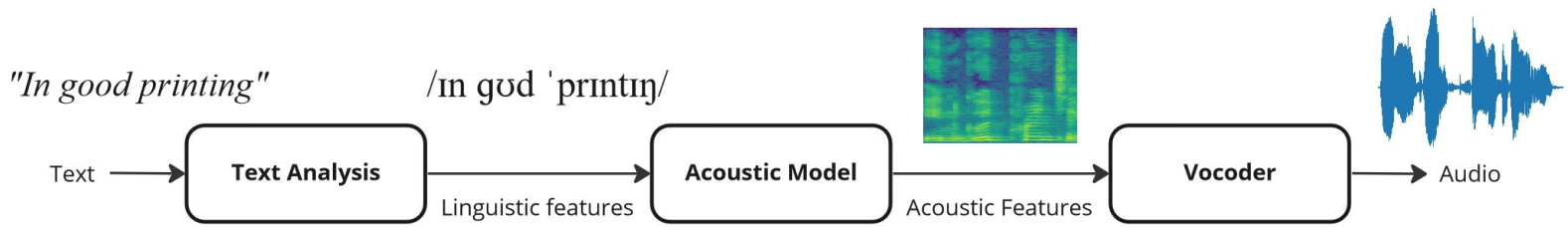
- Problem Introduction
- Past Work
- Problem Statement
- Diffusion Models
- Improvements
- Method & Results
- Discussion & Future Work

Typical TTS Pipeline

Text $\longrightarrow$ Speech

Typical TTS Pipeline

Text $\longrightarrow$ Speech



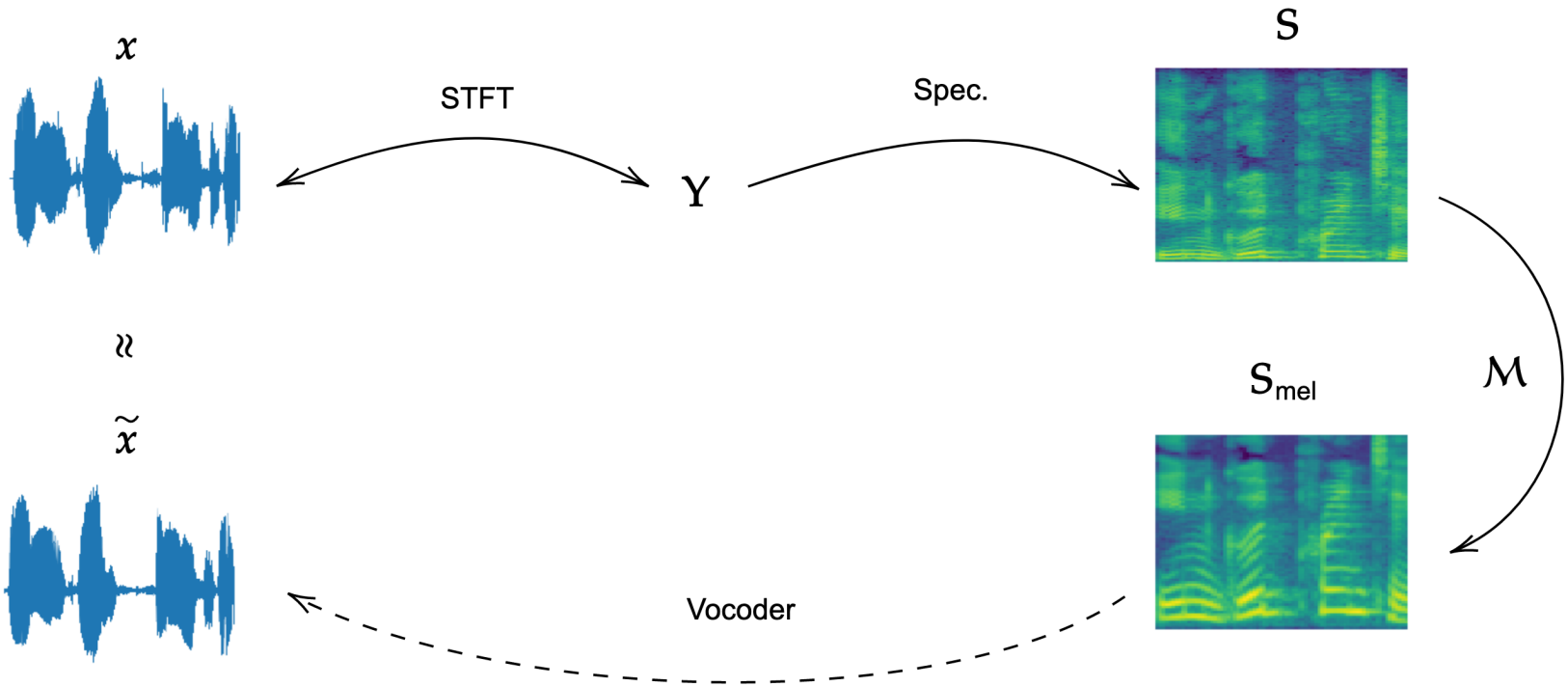
Text Analysis Acoustic Model Vocoder

The Phase Reconstruction Problem

- Goal: Reconstruct signal $\boldsymbol{x}$ from its mel spectrogram $\boldsymbol{S}_{\text{mel}}$

The Phase Reconstruction Problem

- Goal: Reconstruct signal $\boldsymbol{x}$ from its mel spectrogram $\boldsymbol{S}_{\text{mel}}$



The Phase Reconstruction Problem

How has this been done before?

The Phase Reconstruction Problem

How has this been done before?

- Griffin-Lim Reconstruction

The Phase Reconstruction Problem

How has this been done before?

- Griffin-Lim Reconstruction
- Autoregressive Neural Networks (WaveNet)

The Phase Reconstruction Problem

How has this been done before?

- Griffin-Lim Reconstruction
- Autoregressive Neural Networks (WaveNet)
- GANs (HiFi-GAN)

The Phase Reconstruction Problem

How has this been done before?

- Griffin-Lim Reconstruction
- Autoregressive Neural Networks (WaveNet)
- GANs (HiFi-GAN)
- Diffusion (DiffWave)

The Phase Reconstruction Problem

How has this been done before?

- Griffin-Lim Reconstruction
- Autoregressive Neural Networks (WaveNet)
- GANs (HiFi-GAN)
- Diffusion (DiffWave)
- ...and more

Problem Statement

- How does inference speed and generated audio quality compare to a GAN-based vocoder?

Problem Statement

- How does inference speed and generated audio quality compare to a GAN-based vocoder?
- How do the following aspects affect inference speed and audio quality?

Problem Statement

- How does inference speed and generated audio quality compare to a GAN-based vocoder?
- How do the following aspects affect inference speed and audio quality?
  - Variance schedule
  - Time-step importance sampling
  - Noise prior

Problem Statement

- How does inference speed and generated audio quality compare to a GAN-based vocoder?
- How do the following aspects affect inference speed and audio quality?
  - Variance schedule
  - Time-step importance sampling
  - Noise prior
- How does a vocoder trained on audio from one speaker generalize to another?

Problem Statement

- How does inference speed and generated audio quality compare to a GAN-based vocoder?
- How do the following aspects affect inference speed and audio quality?
  - Variance schedule
  - Time-step importance sampling
  - Noise prior
- How does a vocoder trained on audio from one speaker generalize to another?
- Can such a model generate audio faster than real-time?

Diffusion

- How can it be used for vocoding?

Diffusion

- Transform data from q_{data} into a simple distribution p_{latent} ("forward process")

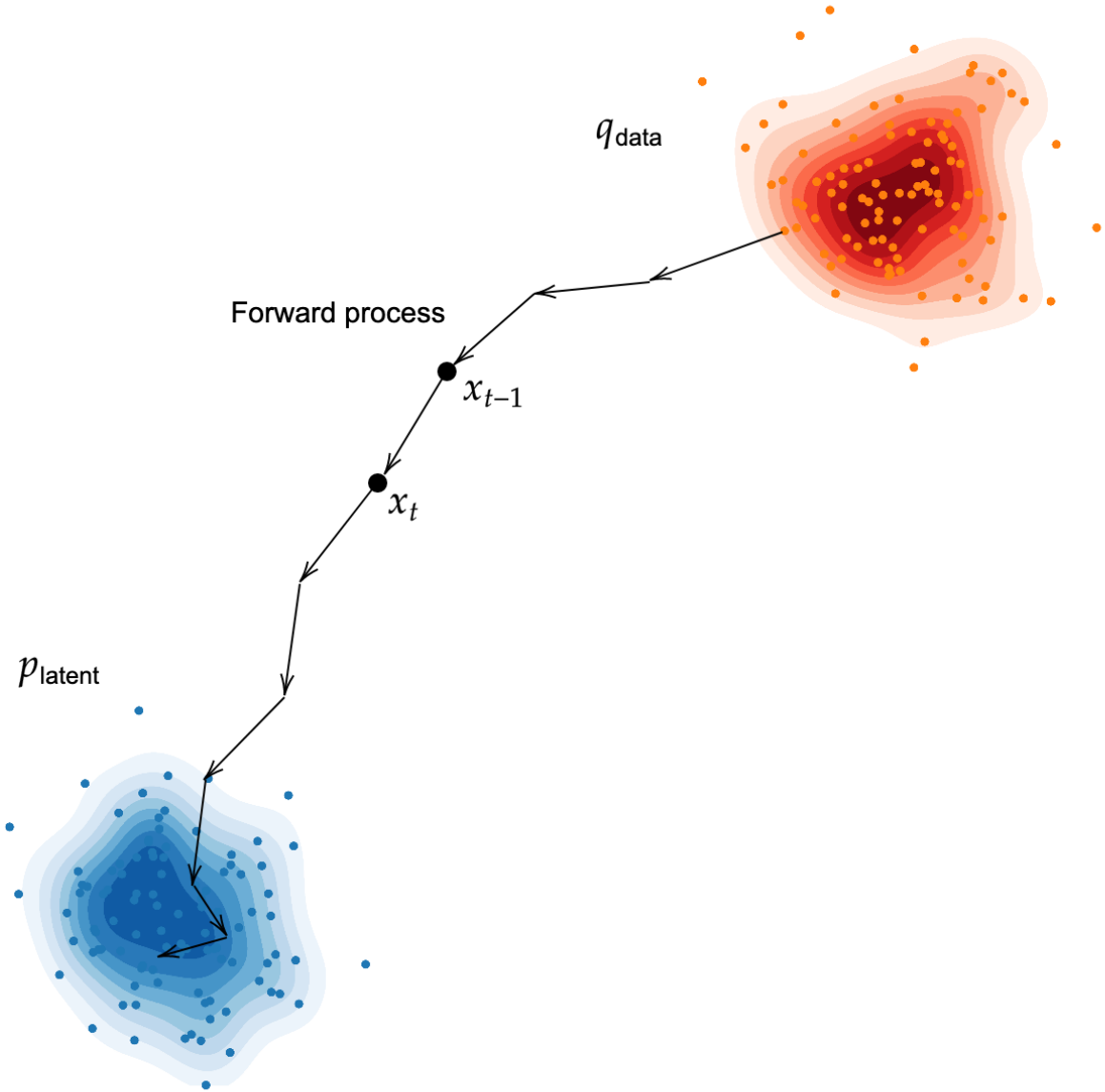
Diffusion

- Transform data from q_{data} into a simple distribution p_{latent} ("forward process")
- Learn to transform data from a simple distribution p_{latent} (e.g. noise) into the complex target distribution q_{data} ("backward process")

Diffusion

- Transform data from q_{data} into a simple distribution p_{latent} ("forward process")
- Learn to transform data from a simple distribution p_{latent} (e.g. noise) into the complex target distribution q_{data} ("backward process")
- Teach a model to perform the inverse transformation in several steps

Forward Process



Forward Process

- Choose a number of diffusion steps T
- Decide how "big" steps we take via a variance schedule β_t
- Each sample is drawn as
  $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t\,; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I})$

Backward Process

- We want to be able to generate samples by reversing the process

Backward Process

- We want to be able to generate samples by reversing the process
- Model each transition as $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, model is to predict the noise added in each step via $\boldsymbol{\varepsilon}_\theta$

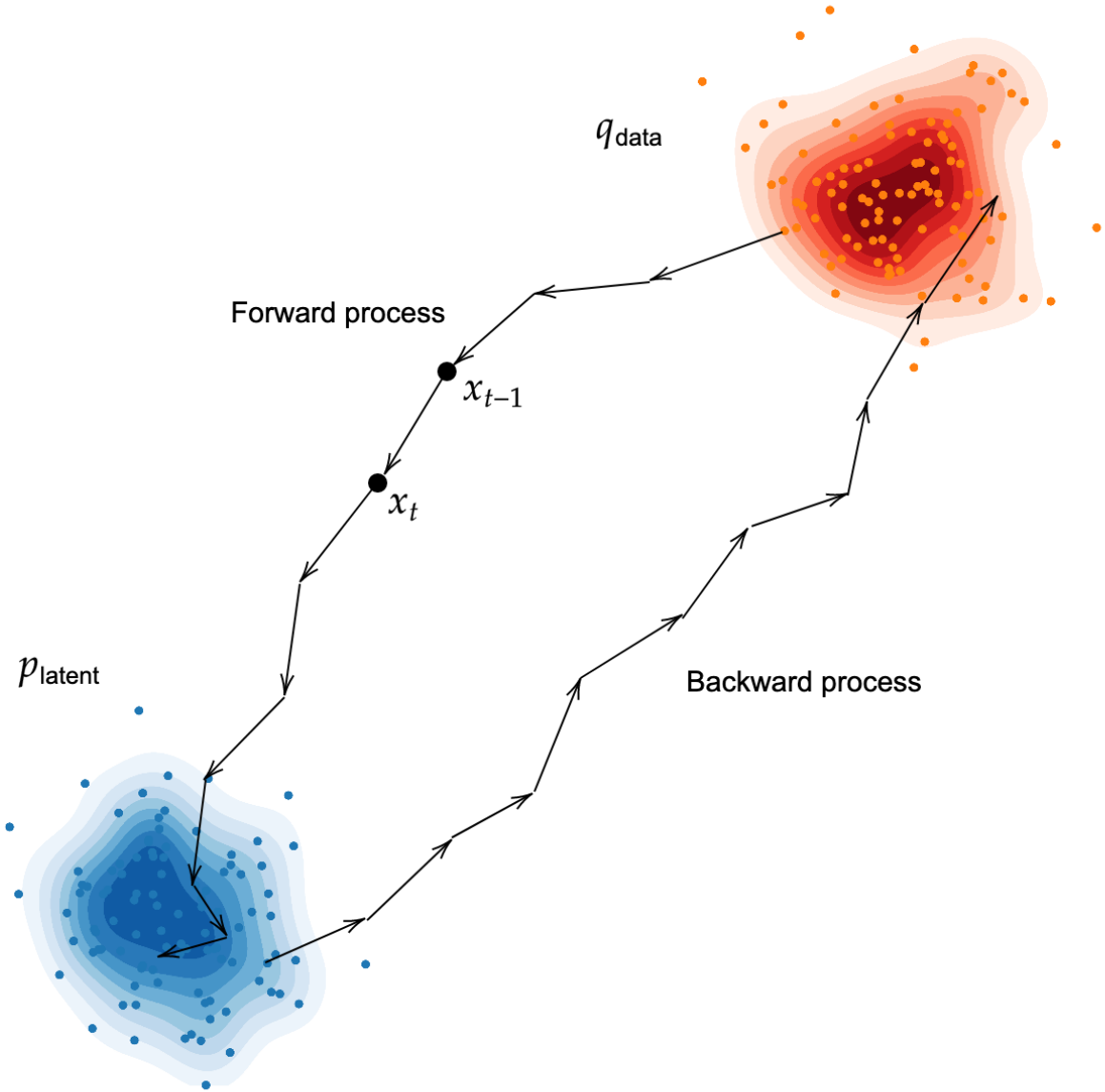
Backward Process

- We want to be able to generate samples by reversing the process
- Model each transition as $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, model is to predict the noise added in each step via $\boldsymbol{\varepsilon}_\theta$
- We condition on more noisy sample $\boldsymbol{x}_t$, diffusion step t, and the mel-spectrogram $\boldsymbol{S}_{\text{mel}}$ in each step

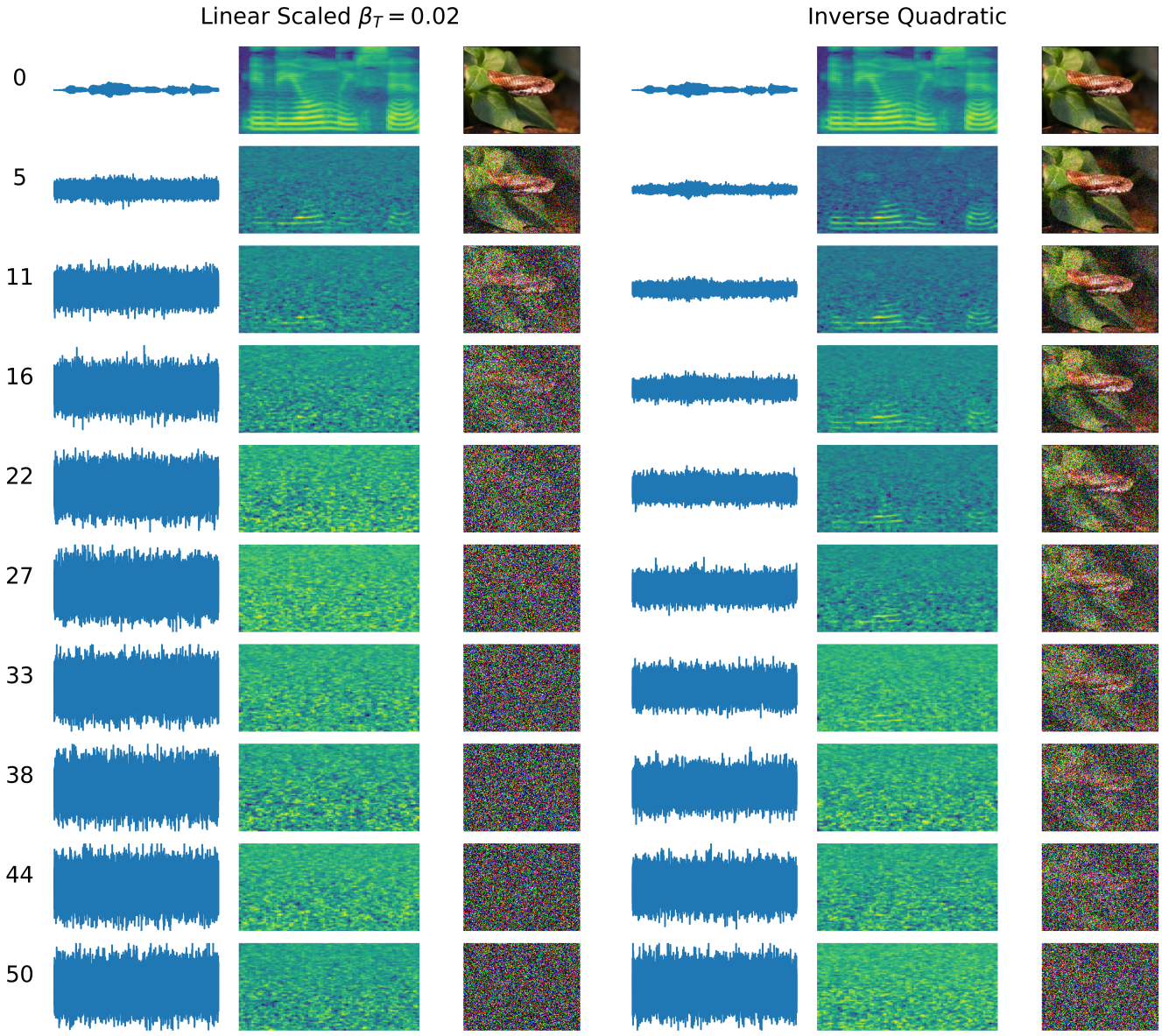
Backward Process

- We want to be able to generate samples by reversing the process
- Model each transition as $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, model is to predict the noise added in each step via $\boldsymbol{\varepsilon}_\theta$
- We condition on more noisy sample $\boldsymbol{x}_t$, diffusion step t, and the mel-spectrogram $\boldsymbol{S}_{\text{mel}}$ in each step
- Training: sample t uniformly over $[1, T]$

Backward Process



Backward Process



Main Considerations

- Choice of variance schedule β_t

Main Considerations

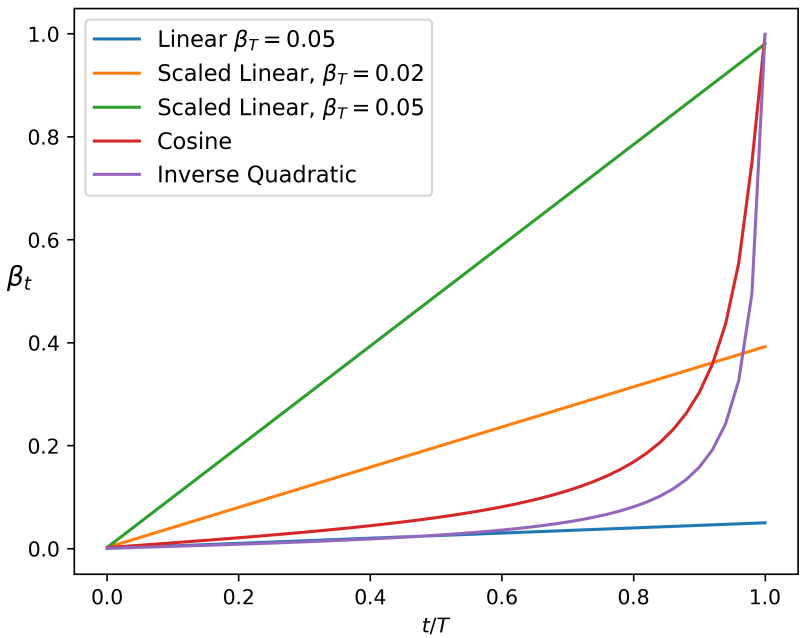
- Choice of variance schedule β_t
- Choice of noise prior p_{latent}

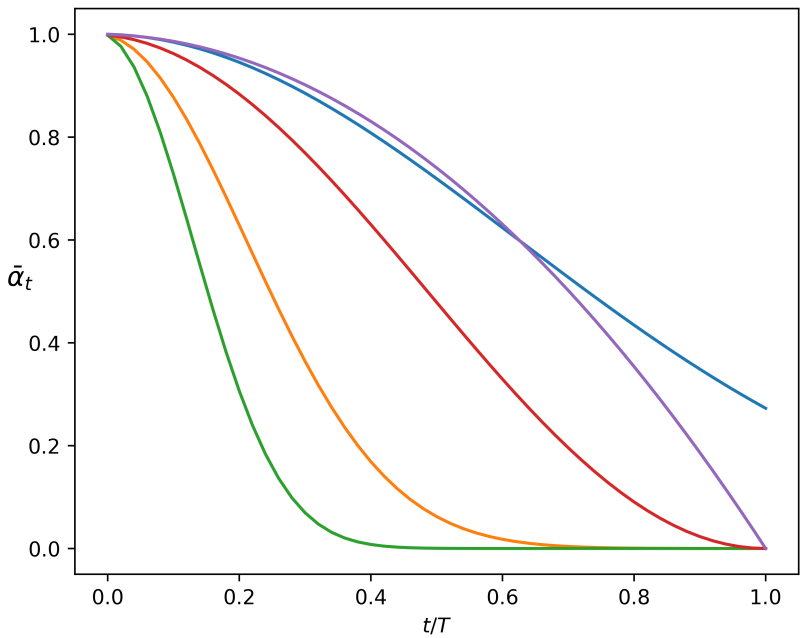
Main Considerations

- Choice of variance schedule β_t
- Choice of noise prior p_{latent}
- Uniform vs importance sampling of t

Variance Schedule

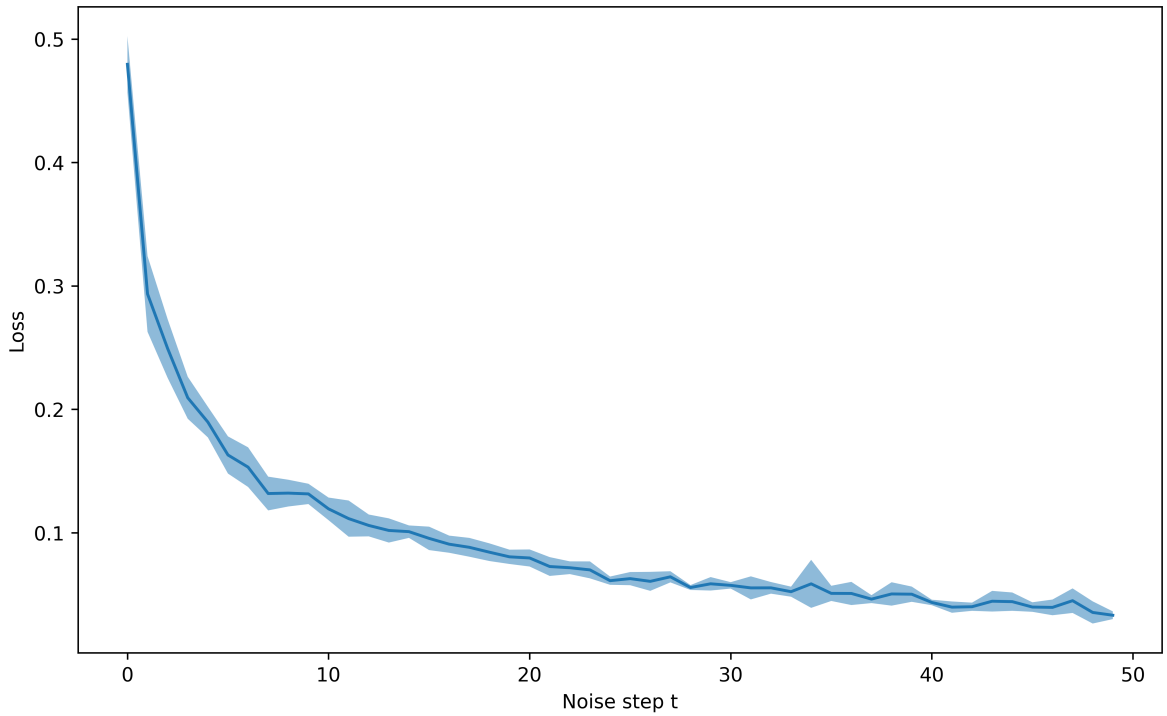
- We do not want to reach white noise too early
- Smooth transition from data to noise





Importance Sampling

Gradient is noisy when training! Most likely from a skewed loss:



Hypothesis: some steps are harder to learn than others

Idea: sample t weighted by the loss

Choice of Prior

Instead of starting from unit white noise, can we choose a better prior?

Choice of Prior

Instead of starting from unit white noise, can we choose a better prior?

- Use information from $\boldsymbol{S}_{\text{mel}}$ to choose a prior

Choice of Prior

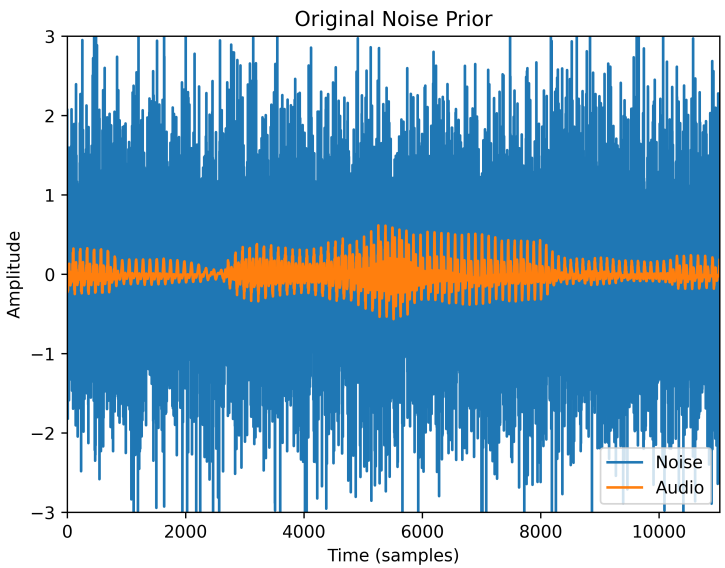
Instead of starting from unit white noise, can we choose a better prior?

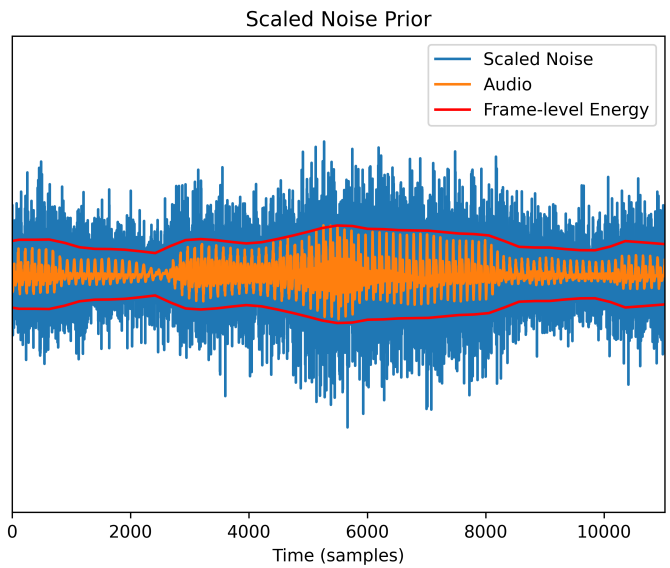
- Use information from $\boldsymbol{S}_{\text{mel}}$ to choose a prior
- Get variance of starting noise from frame-level energy of $\boldsymbol{S}_{\text{mel}}$

Choice of Prior

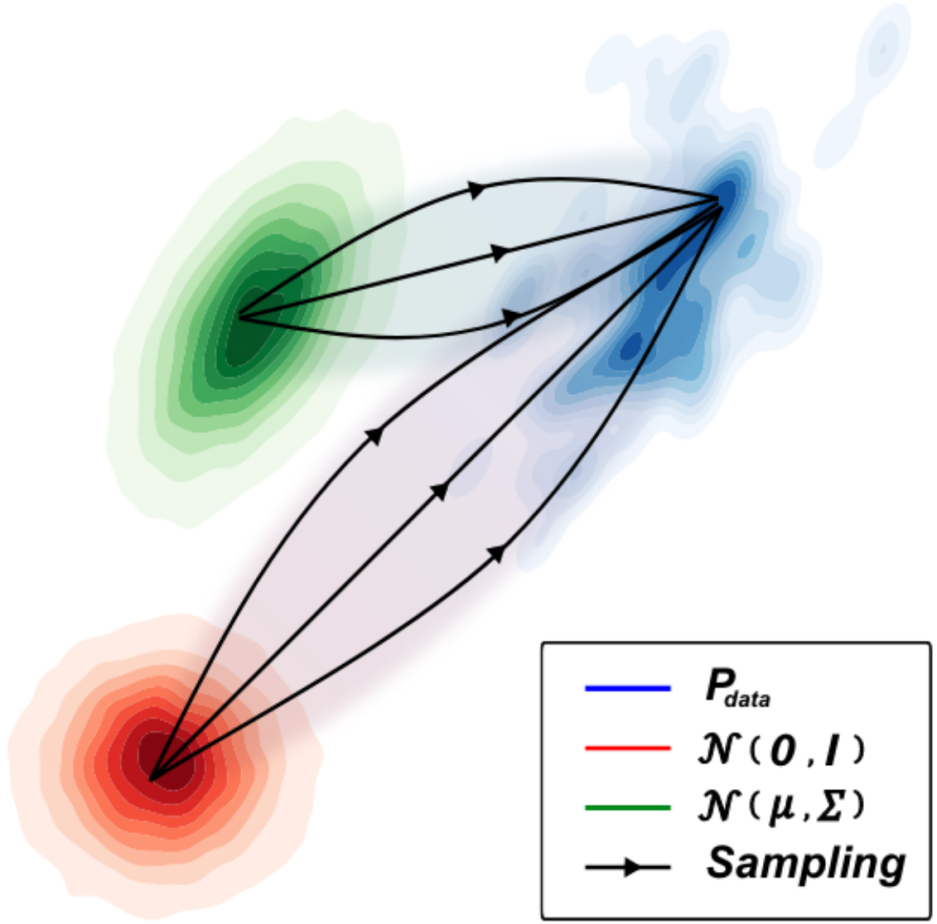
Instead of starting from unit white noise, can we choose a better prior?

- Use information from $\boldsymbol{S}_{\text{mel}}$ to choose a prior
- Get variance of starting noise from frame-level energy of $\boldsymbol{S}_{\text{mel}}$





Choice of Prior



Metrics

- Approximate Mean Opinion Score (AMOS)
  - Scale 1-5, finetuned audio model
- Log-mel Spectrogram Mean Absolute Error (LS-MAE)
- Peak Signal-to-Noise Ratio (PSNR)
- Multi-resolution STFT Error (MRSE)

Data

- LJ Speech
  - Training set
  - 24 hours of speech from audiobooks
  - 13 100 audio files, 95 in test set
  - Sample rate: 22 050 Hz
- LibriTTS
  - Test set of other speaker
  - 515 audio files

Model

- DiffWave, based on dilated convolutions
- 2.6M parameters, $T = 50$
- Short inference schedule with $T = 6$

Method & Results

Experiments

- Variance Schedules
- Time-step Sampling
- Noise Prior
- Model Size
- Inference Speed
- Longer Training

Method & Results: Variance Schedule

Objective Metrics:

	LJ-test			Libri-test		
Model	**LS-MAE ↓**	**PSNR ↑**	**MRSE ↓**	**LS-MAE ↓**	**PSNR ↑**	**MRSE ↓**
Griffin-Lim	0.302	25.517	1.131	0.227	27.011	1.027
Linear $\beta_T = 0.05$	0.550	20.331	1.279	0.680	23.510	1.341
Scaled Linear $\beta_T = 0.05$	0.779	17.340	1.593	0.818	20.510	1.522
Scaled Linear $\beta_T = 0.02$	0.727	18.077	1.511	0.812	21.056	1.483
Cosine	0.699	17.829	1.512	0.712	20.982	1.463
Inverse Quadratic	0.541	20.190	1.270	0.698	23.524	1.345

Method & Results: Variance Schedule

Subjective Metrics:

Model	**AMOS** ↑ (LJ-test)	**AMOS** ↑ (Libri-test)
Ground Truth	3.63 ± 0.26	2.68 ± 0.38
Griffin-Lim	1.57 ± 0.15	1.35 ± 0.18
Linear $\beta_T = 0.05$	2.88 ± 0.23	2.14 ± 0.26
Scaled Linear $\beta_T = 0.05$	2.27 ± 0.17	1.77 ± 0.26
Scaled Linear $\beta_T = 0.02$	2.40 ± 0.21	1.82 ± 0.25
Cosine	2.39 ± 0.19	1.83 ± 0.27
Inverse quadratic	2.89 ± 0.23	2.17 ± 0.27

Method & Results: Variance Schedule

- Original Linear and Inverse Quadratic best

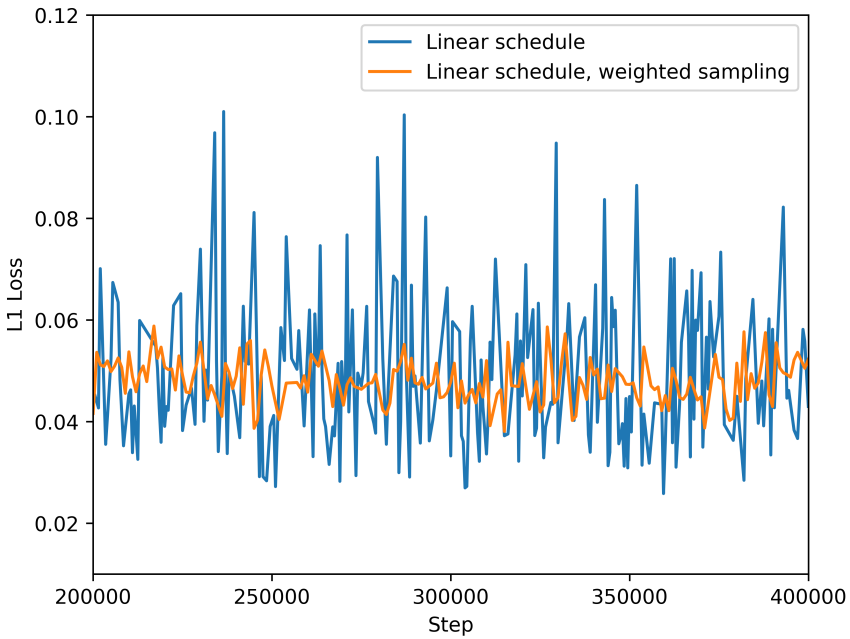
Method & Results: Variance Schedule

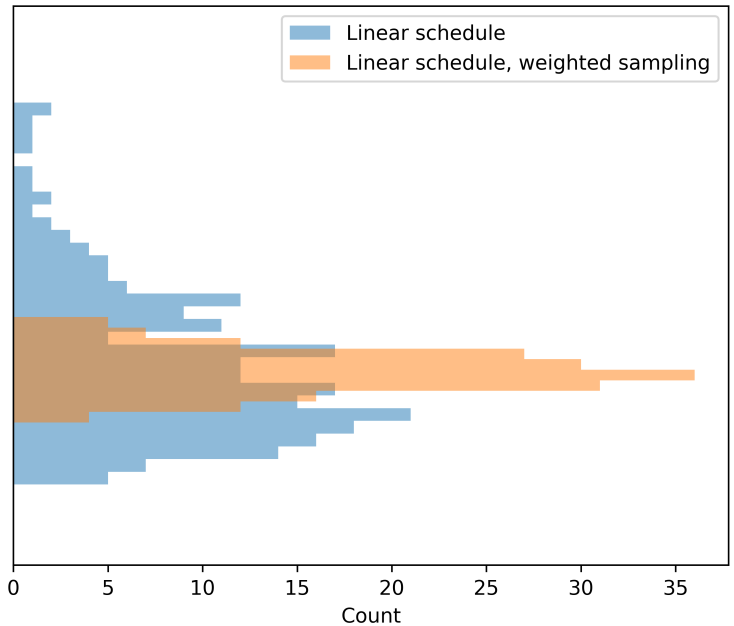
- Original Linear and Inverse Quadratic best
- Scaled surprisingly bad

Method & Results: Variance Schedule

- Original Linear and Inverse Quadratic best
- Scaled surprisingly bad
- Griffin Lim best objective metrics, worst AMOS

Method & Results: Time-step Sampling





Method & Results: Time-step Sampling

Objective Metrics:

	LJ-test			Libri-test		
Model	**LS-MAE ↓**	**PSNR ↑**	**MRSE ↓**	**LS-MAE ↓**	**PSNR ↑**	**MRSE ↓**
Uniform sampling	0.550	20.331	1.279	0.680	23.510	1.341
Uniform sampling (fast)	0.562	20.069	1.314	0.684	23.461	1.379
Importance sampling	0.519	19.494	1.285	0.605	23.485	1.287
Importance sampling (fast)	0.518	19.333	1.318	0.587	23.318	1.309

Method & Results: Time-step Sampling

Subjective Metrics:

Model	**AMOS** ↑ (LJ-test)	**AMOS** ↑ (Libri-test)
Ground truth	3.63 ± 0.26	2.68 ± 0.38
Uniform sampling	2.88 ± 0.23	2.14 ± 0.26
Uniform sampling (fast)	2.75 ± 0.24	2.07 ± 0.25
Importance sampling	2.87 ± 0.25	2.15 ± 0.28
Importance sampling (fast)	2.78 ± 0.24	2.09 ± 0.27

Method & Results: Time-step Sampling

- Loss variance reduced

Method & Results: Time-step Sampling

- Loss variance reduced
- Both objective & subjective scores similar

Method & Results: Time-step Sampling

- Loss variance reduced
- Both objective & subjective scores similar
- Not significantly more robust to shorter schedule

Method & Results: Noise Prior

Objective Metrics:

	LJ-test			Libri-test		
Model	**LS-MAE ↓**	**PSNR ↑**	**MRSE ↓**	**LS-MAE ↓**	**PSNR ↑**	**MRSE ↓**
DiffWave	0.550	20.331	1.279	0.680	23.510	1.341
DiffWave (fast)	0.562	20.069	1.314	0.684	23.461	1.379
PriorGrad	0.474	27.499	1.343	0.615	24.731	1.564
PriorGrad (fast)	0.455	27.236	1.336	0.567	24.798	1.518

Method & Results: Noise Prior

Subjective Metrics:

Model	**AMOS** ↑ (LJ-test)	**AMOS** ↑ (Libri-test)
Ground truth	3.63 ± 0.26	2.68 ± 0.38
DiffWave	2.88 ± 0.23	2.14 ± 0.26
DiffWave (fast)	2.75 ± 0.24	2.07 ± 0.25
PriorGrad	3.00 ± 0.29	2.19 ± 0.33
PriorGrad (fast)	2.74 ± 0.29	2.00 ± 0.28

Method & Results: Noise Prior

- Both objective and subjective scores improved

Method & Results: Noise Prior

- Both objective and subjective scores improved
- Slightly worse performance for short schedule

Method & Results: Noise Prior

- Both objective and subjective scores improved
- Slightly worse performance for short schedule
- Higher MRSE might indicate overfitting

Method & Results: Model Size

How does model size affect performance?

Objective Metrics:

	LJ-test			Libri-test		
Model	**LS-MAE ↓**	**PSNR ↑**	**MRSE ↓**	**LS-MAE ↓**	**PSNR ↑**	**MRSE ↓**
Base	0.550	20.331	1.279	0.680	23.510	1.341
Base + IQ + IS	0.616	19.154	1.381	0.745	23.197	1.389
Base (fast)	0.562	20.069	1.314	0.684	23.461	1.379
Base + IQ + IS (fast)	0.634	18.800	1.421	0.766	23.145	1.437
Small	0.547	19.860	1.311	0.692	22.752	1.409
Small + IQ + IS	0.611	18.945	1.330	0.736	23.170	1.392
Small (fast)	0.535	19.678	1.342	0.691	22.787	1.408
Small + IQ + IS (fast)	0.615	18.907	1.358	0.730	23.160	1.420

Method & Results: Model Size

Subjective Metrics:

Model	**RTF** ↓ (CPU)	**RTF** ↓ (GPU)	**AMOS** ↑ (LJ-test)	**AMOS** ↑ (Libri-test)
Base	44.38	0.57	2.88 ± 0.23	2.14 ± 0.26
Base + IQ + IS	48.61	0.56	2.74 ± 0.23	2.12 ± 0.26
Base (fast)	5.26	0.061	2.75 ± 0.24	2.07 ± 0.25
Base + IQ + IS (fast)	5.44	0.061	2.68 ± 0.22	2.06 ± 0.25
Small	33.07	0.43	2.76 ± 0.22	2.07 ± 0.26
Small + IQ + IS	33.28	0.41	2.76 ± 0.21	2.09 ± 0.27
Small (fast)	4.04	0.046	2.65 ± 0.22	2.07 ± 0.27
Small + IQ + IS (fast)	4.10	0.046	2.63 ± 0.22	2.00 ± 0.25

Method & Results: Model Size

- 70% model size, only 75% of inference time

Method & Results: Model Size

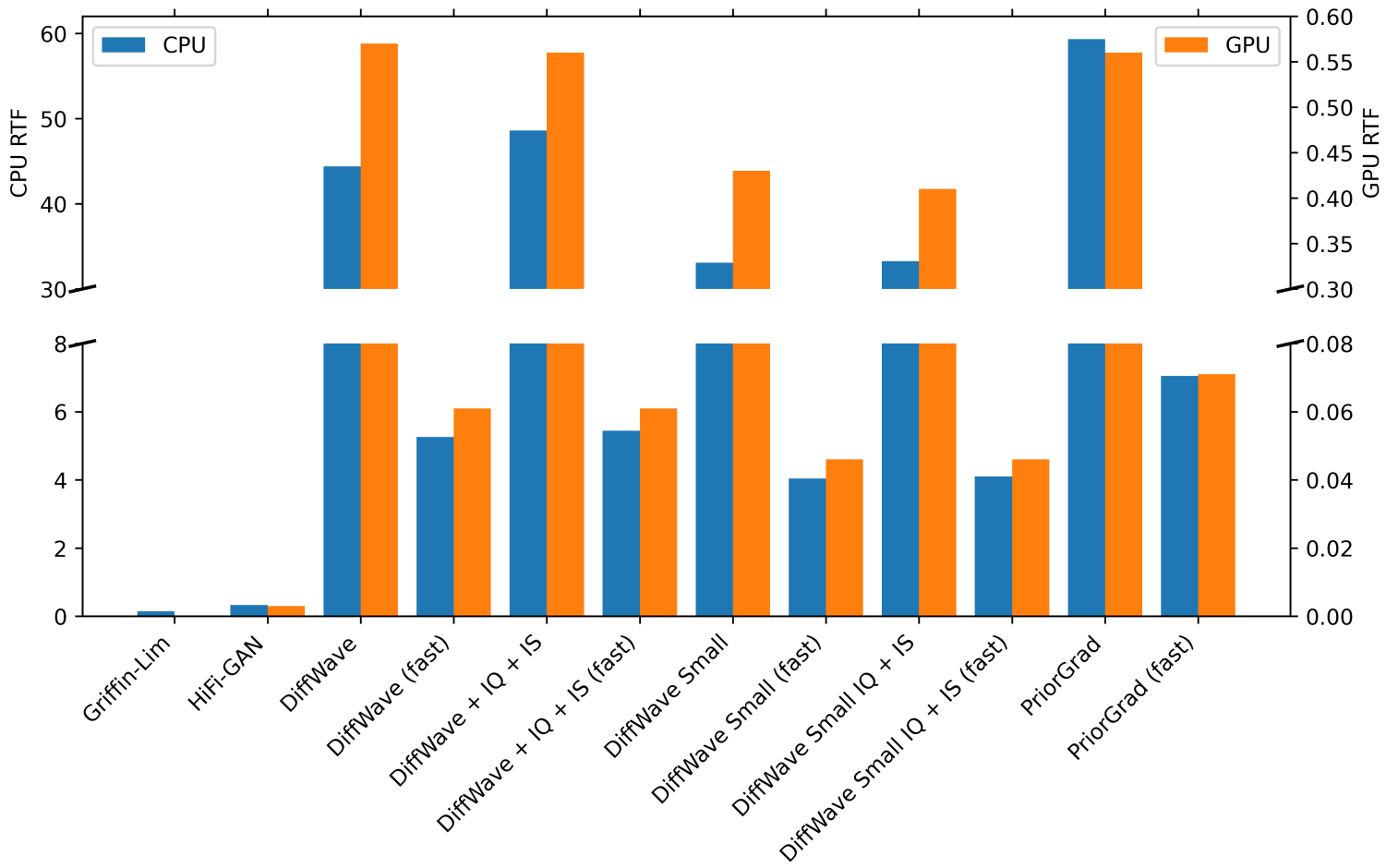
- 70% model size, only 75% of inference time
- Minimal difference in audio quality

Method & Results: Model Size

- 70% model size, only 75% of inference time
- Minimal difference in audio quality
- IQ + IS did not result in a better smaller model

Method & Results: Inference Speed

How do the models compare in terms of inference speed?



Method & Results: Inference Speed

- Slowest: PriorGrad, 60x slower than real-time on CPU, 0.55x on GPU

Method & Results: Inference Speed

- Slowest: PriorGrad, 60x slower than real-time on CPU, 0.55x on GPU
- Fastest: Small Diffwave (fast), 4x slower than real-time on CPU, 0.05x on GPU

Method & Results: Inference Speed

- Slowest: PriorGrad, 60x slower than real-time on CPU, 0.55x on GPU
- Fastest: Small Diffwave (fast), 4x slower than real-time on CPU, 0.05x on GPU
- Fast sampling: 12% of steps, 9x speedup

Method & Results: Inference Speed

- Slowest: PriorGrad, 60x slower than real-time on CPU, 0.55x on GPU
- Fastest: Small Diffwave (fast), 4x slower than real-time on CPU, 0.05x on GPU
- Fast sampling: 12% of steps, 9x speedup
- Still 15 times slower than HiFi-GAN

Method & Results: Longer Training

How does longer training affect performance?

Objective Metrics:

	LJ-test			Libri-test		
Model	**LS-MAE**	**PSNR**	**MRSE**	**LS-MAE**	**PSNR**	**MRSE**
HiFi-GAN (2.5M)	0.345	23.854	1.236	0.261	27.673	1.155
DiffWave (500k)	0.550	20.331	1.279	0.680	23.510	1.341
DiffWave (2.5M)	0.568	21.049	1.250	0.759	23.678	1.370
DiffWave (2.5M) (fast)	0.561	20.963	1.259	0.745	23.745	1.383
DiffWave + IQ + IS (2.5M)	0.577	21.282	1.252	0.791	23.754	1.379
DiffWave + IQ + IS (2.5M) (fast)	0.576	21.143	1.271	0.781	23.822	1.392

Method & Results: Longer Training

Subjective Metrics:

Model	**AMOS** (LJ-test)	**AMOS** (Libri-test)
Ground truth	3.63 ± 0.26	2.68 ± 0.38
HiFi-GAN (2.5M)	2.70 ± 0.25	2.29 ± 0.38
DiffWave (500K)	2.88 ± 0.23	2.14 ± 0.26
DiffWave (2.5M)	3.05 ± 0.25	2.31 ± 0.30
DiffWave (2.5M) (fast)	2.99 ± 0.23	2.24 ± 0.29
DiffWave + IQ + IS (2.5M)	3.05 ± 0.25	2.30 ± 0.29
DiffWave + IQ + IS (2.5M) (fast)	2.98 ± 0.42	2.23 ± 0.27

Method & Results: Longer Training

- 5x longer training, only slightly better scores

Method & Results: Longer Training

- 5x longer training, only slightly better scores
- Similar quality to HiFi-GAN

Method & Results: Longer Training

- 5x longer training, only slightly better scores
- Similar quality to HiFi-GAN
- IQ + IS resulted in similar scores to base

Method & Results: Recap

- Variance Schedule: Original Linear & IQ similar, image generation schedules worse

Method & Results: Recap

- Variance Schedule: Original Linear & IQ similar, image generation schedules worse
- Time-step Sampling: Lower variance, no change in performance

Method & Results: Recap

- Variance Schedule: Original Linear & IQ similar, image generation schedules worse
- Time-step Sampling: Lower variance, no change in performance
- Noise Prior: Improved performance

Method & Results: Recap

- Variance Schedule: Original Linear & IQ similar, image generation schedules worse
- Time-step Sampling: Lower variance, no change in performance
- Noise Prior: Improved performance
- Smaller Model: slightly faster, quality approximately the same

Method & Results: Recap

- Variance Schedule: Original Linear & IQ similar, image generation schedules worse
- Time-step Sampling: Lower variance, no change in performance
- Noise Prior: Improved performance
- Smaller Model: slightly faster, quality approximately the same
- Inference Speed: Fastest 0.05x RTF on GPU, 4x on CPU

Method & Results: Recap

- Variance Schedule: Original Linear & IQ similar, image generation schedules worse
- Time-step Sampling: Lower variance, no change in performance
- Noise Prior: Improved performance
- Smaller Model: slightly faster, quality approximately the same
- Inference Speed: Fastest 0.05x RTF on GPU, 4x on CPU
- Longer Training: 5x longer training, similar scores

Discussion

Main takeaways:

- Possible to perform vocoding using diffusion on GPU, slow on CPU, further speedup possible

Discussion

Main takeaways:

- Possible to perform vocoding using diffusion on GPU, slow on CPU, further speedup possible
- Choice of schedule important, should not reach noise too early

Discussion

Main takeaways:

- Possible to perform vocoding using diffusion on GPU, slow on CPU, further speedup possible
- Choice of schedule important, should not reach noise too early
- Importance sampling: not more robust, but better variance

Discussion

Main takeaways:

- Possible to perform vocoding using diffusion on GPU, slow on CPU, further speedup possible
- Choice of schedule important, should not reach noise too early
- Importance sampling: not more robust, but better variance
- Able to generalize to another speaker, AMOS hard to interpret

Discussion

Main takeaways:

- Possible to perform vocoding using diffusion on GPU, slow on CPU, further speedup possible
- Choice of schedule important, should not reach noise too early
- Importance sampling: not more robust, but better variance
- Able to generalize to another speaker, AMOS hard to interpret
- Objective Metrics not always reliable

Discussion

Main takeaways:

- Possible to perform vocoding using diffusion on GPU, slow on CPU, further speedup possible
- Choice of schedule important, should not reach noise too early
- Importance sampling: not more robust, but better variance
- Able to generalize to another speaker, AMOS hard to interpret
- Objective Metrics not always reliable
- Similar quality to HiFi-GAN, but slower

Future Work

- IQ + IS + PriorGrad

Future Work

- IQ + IS + PriorGrad
- Better theoretical understanding of the schedule is needed

Future Work

- IQ + IS + PriorGrad
- Better theoretical understanding of the schedule is needed
- Learnable $\boldsymbol{\Sigma}_\theta$, condition on $\bar{\alpha}_t$

Future Work

- IQ + IS + PriorGrad
- Better theoretical understanding of the schedule is needed
- Learnable $\boldsymbol{\Sigma}_\theta$, condition on $\bar{\alpha}_t$
- Smaller model, shorter schedule

Future Work

- IQ + IS + PriorGrad
- Better theoretical understanding of the schedule is needed
- Learnable $\boldsymbol{\Sigma}_\theta$, condition on $\bar{\alpha}_t$
- Smaller model, shorter schedule
- Better metrics

Future Work

- IQ + IS + PriorGrad
- Better theoretical understanding of the schedule is needed
- Learnable $\boldsymbol{\Sigma}_\theta$, condition on $\bar{\alpha}_t$
- Smaller model, shorter schedule
- Better metrics
- Higher sample rate, Distillation, Super-resolution

Thank you!