Role of Samplers in Diffusion Based Generative Models

Ernests Lavrinovics
MSc Medialogy MED10
Aalborg University, Copenhagen

Agenda

- Introduction
  - Generative AI and Diffusion-Based Models
  - Applications
- Background
  - Diffusion-Based Speech Enhancement
- Problem Statement
- Methodology
- Results
- Summary

Introduction

Generative AI Models

- Discriminative AI – learn the decision boundary
  - Conv.Neural Nets (CNN), Multilayer Percept. (MLP), Transformers
- Generative AI – learn the underlying data distribution
  - Variational Autoencoders (VAE), Generative Adversarial Networks (GAN), Boltzmann machines, Transformers, Diffusion models
- Dependent on the objective and training setup

Image from https://github.com/NVlabs/stylegan

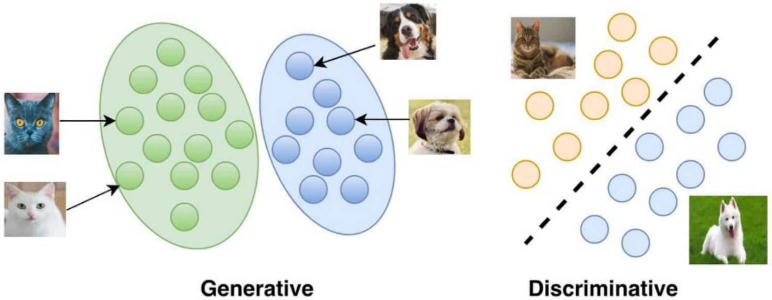
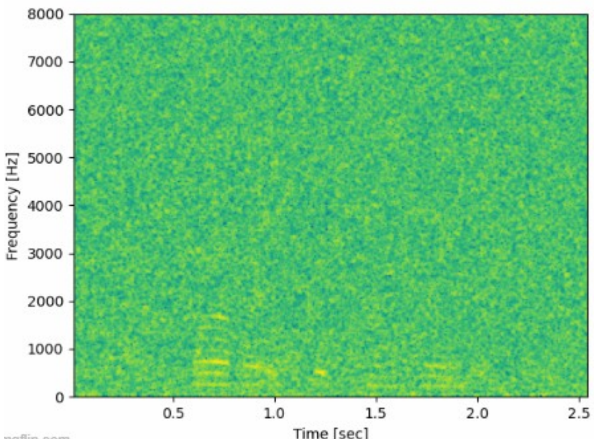


Image from https://vitalflux.com/generative-vs-discriminative-models-examples/

Introduction

Diffusion-Based Generative AI Models

- Diffusion (iterative process)
  - Forward-diffusion: corrupt a datapoint with noise
  - Reverse-diffusion: reverse the noise into data
- This process can be optionally be guided via conditioning



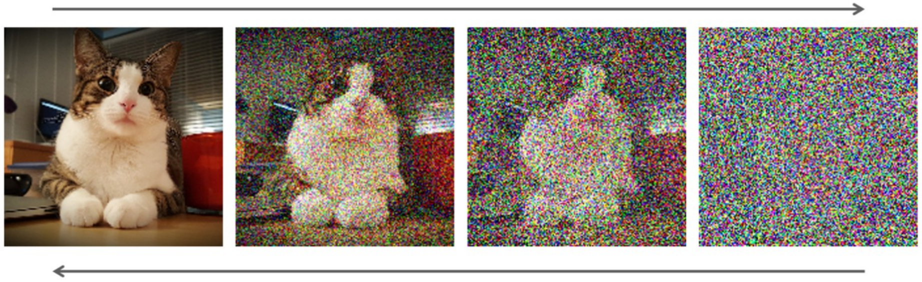


Image from https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-2/

Introduction

Applications of Diffusion Models

- Image processing
  - Restoration (super resolution, inpainting, colorization)
  - Anomaly detection, semantic segmentation
- Audio processing
  - Speech and music enhancement/generation (communications, restoration)
- Dataset generation

OpenAI's Dall-E 2

Left: "Panda mad scientist mixing sparkling chemicals"
Right: "A corgi's head depicted as an explosion in nebula"

Noisy / Enhanced

"We'll eat frozen pizzas all day.. All day every day"
UNIVERSE: https://serrjoa.github.io/projects/universe/

Background

Diffusion-Based Speech Enhancement

- Task is **conditional generation**, we want to guide the process based on an impaired speech signal
- Ref. contribution* takes theory from image processing
- Forward-process expressed using an SDE

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, \mathbf{y})\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

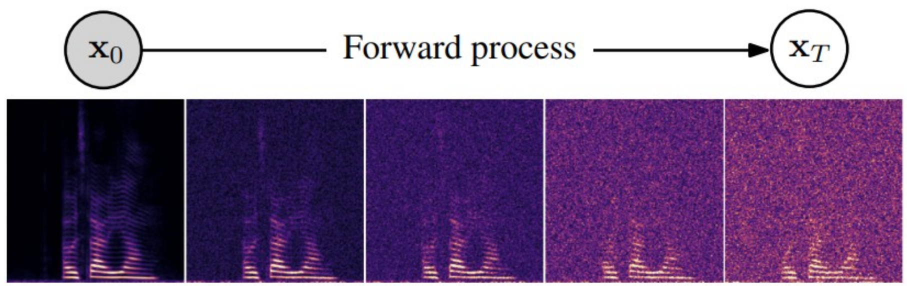


Image from*

*Richter, et.al. "Speech Enhancement and Dereverberation with Diffusion-Based Generative Models", IEEE/ACM Transactions on Audio, Speech, and Language Processing

Background

Diffusion-Based Speech Enhancement: Forward Process

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, \mathbf{y})\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

$$\mathbf{f}(\mathbf{x}_t, \mathbf{y}) := \gamma(\mathbf{y} - \mathbf{x}_t)$$ // y = noisy, x_0 = clean, γ = *stiffness scalar*

$$g(t) := \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t \sqrt{2 \log \left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)}$$ // set of *dw* magnitudes

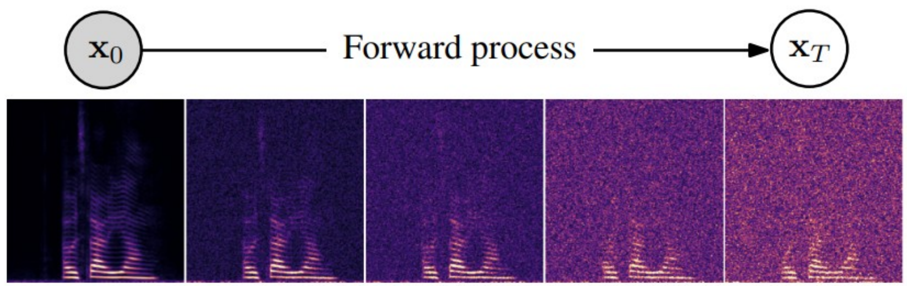


Image from*

*Richter, et.al. "Speech Enhancement and Dereverberation with Diffusion-Based Generative Models", IEEE/ACM Transactions on Audio, Speech, and Language Processing

Background

Diffusion-Based Speech Enhancement: Reverse Process

- The reverse process is also called ***sampling***

$$\mathrm{d}\mathbf{x}_t = \left[-\mathbf{f}(\mathbf{x}_t, \mathbf{y}) + g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$$

- Predict by integrating reverse time SDE with an SDE solver
- Correct by numerical optimization

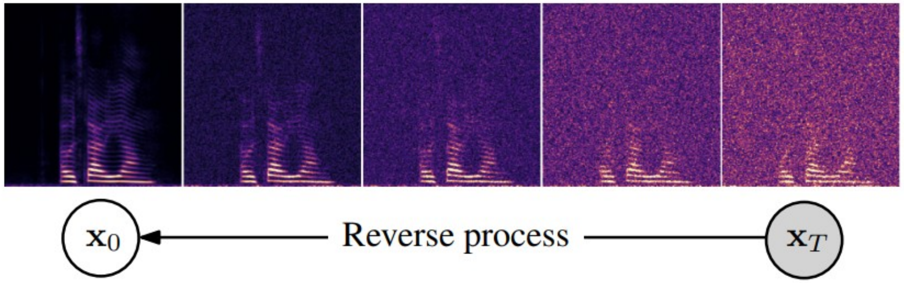


Image from*

*Richter, et.al. "Speech Enhancement and Dereverberation with Diffusion-Based Generative Models", IEEE/ACM Transactions on Audio, Speech, and Language Processing

Background

Diffusion-Based Speech Enhancement: Reverse Process

$$\mathrm{d}\mathbf{x}_t = \left[-\mathbf{f}(\mathbf{x}_t, \mathbf{y}) + g(t)^2 \underline{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})}\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$$

$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$ // Gradient of log-probability density w/r to data

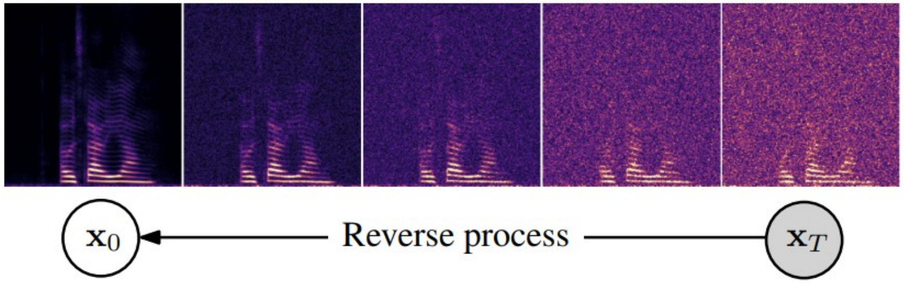


Image from*

*Richter, et.al. "Speech Enhancement and Dereverberation with Diffusion-Based Generative Models", IEEE/ACM Transactions on Audio, Speech, and Language Processing

Background

Diffusion-Based Speech Enhancement: Reverse Process

$$\mathrm{d}\mathbf{x}_t = \left[-\mathbf{f}(\mathbf{x}_t, \mathbf{y}) + g(t)^2 \underline{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})}\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$$

$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$ // Gradient of log-probability density w/r to data

$\mathrm{d}\mathbf{x}_t = \left[-\mathbf{f}(\mathbf{x}_t, \mathbf{y}) + g(t)^2 \mathbf{s}_\theta(\mathbf{x}_t, \mathbf{y}, t)\right] \mathrm{d}t$ // In practice a parametrized neural net sθ

In the ideal scenario sθ $= \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$

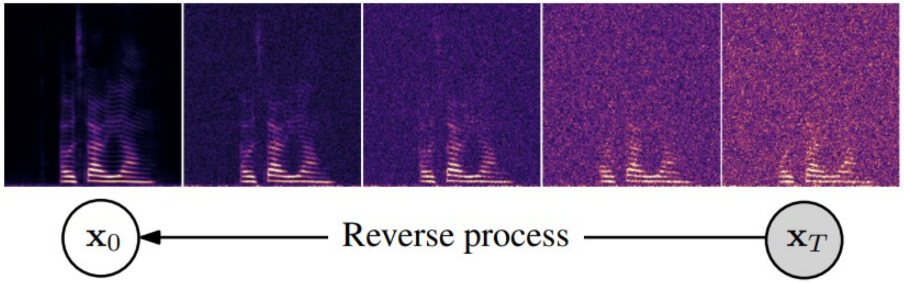


Image from*

*Richter, et.al. "Speech Enhancement and Dereverberation with Diffusion-Based Generative Models", IEEE/ACM Transactions on Audio, Speech, and Language Processing

Background

Diffusion-Based Speech Enhancement: Noise Schedule

- ***σ*** controls the noise injections in forward/reverse processes within a defined min/max range

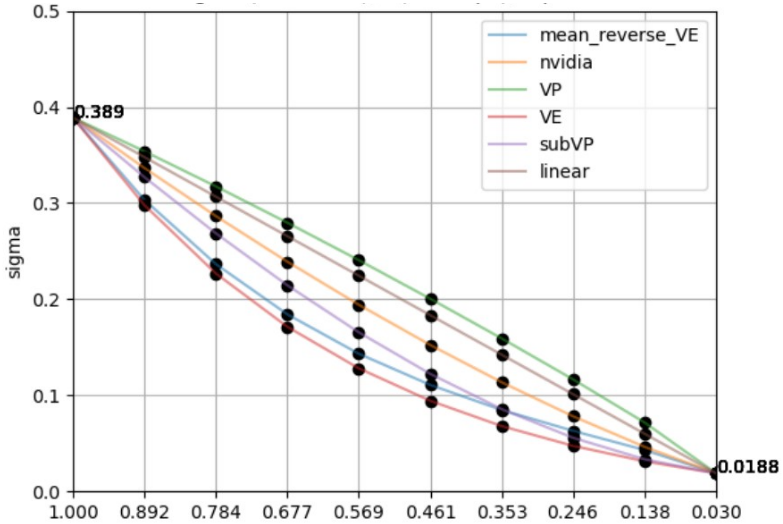
$$\sigma(t)^2 = \frac{\sigma_{\min}^2 \left(\left({}^{\sigma_{\max}}/_{\sigma_{\min}} \right)^{2t} - \mathrm{e}^{-2\gamma t} \right) \log\left({}^{\sigma_{\max}}/_{\sigma_{\min}} \right)}{\gamma + \log\left({}^{\sigma_{\max}}/_{\sigma_{\min}} \right)}$$

- Used during
  - Training: $\arg\min_{\theta} \mathbb{E}_{t,(\mathbf{x}_0,\mathbf{y}),\mathbf{z},\mathbf{x}_t|(\mathbf{x}_0,\mathbf{y})} \left[\left\| \mathbf{s}_\theta(\mathbf{x}_t, \mathbf{y}, t) + \frac{\mathbf{z}}{\sigma(t)} \right\|_2^2 \right]$
  - Sampling: Corrector algorithm (score and Gaussian noise scaling)

Background

Diffusion-Based Speech Enhancement: Noise Schedule

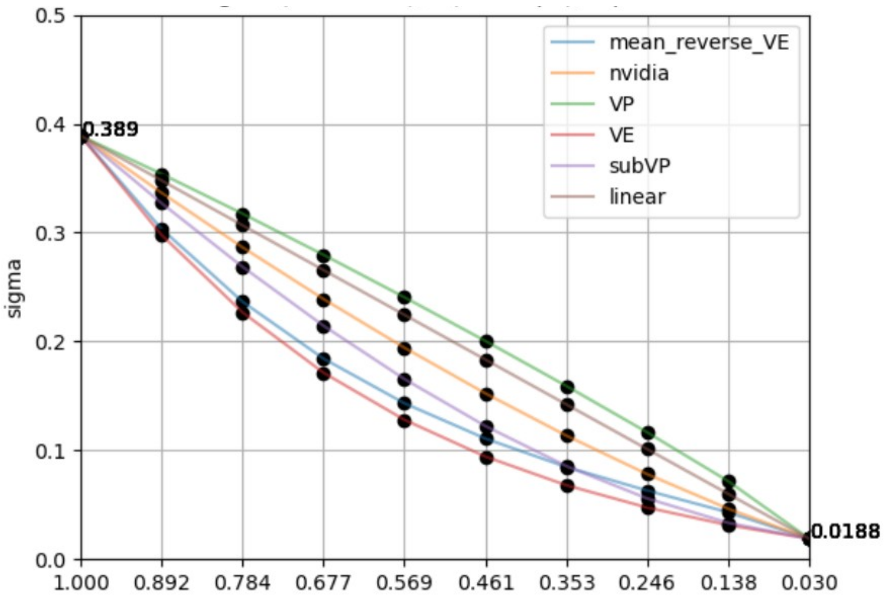
- Sampling routines in production do not need to follow training [2]
- Schedules can be fine-tuned during inference and is crucial for performance [1, 2]



[1] Chen T., On the importance of noise scheduling for diffusion models. arXiv preprint arXiv:2301.10972, 2023
[2] Karras T., et.al.. Elucidating the design space of diffusion-based generative models. NeurIPS, 2022

Problem Statement

- Diffusion models have flexible, tunable moving components
  - Component of the sampling routine – noise schedule
- Investigate the interplay generated output and the noise scheduler
- Empirical study through a set of experiments

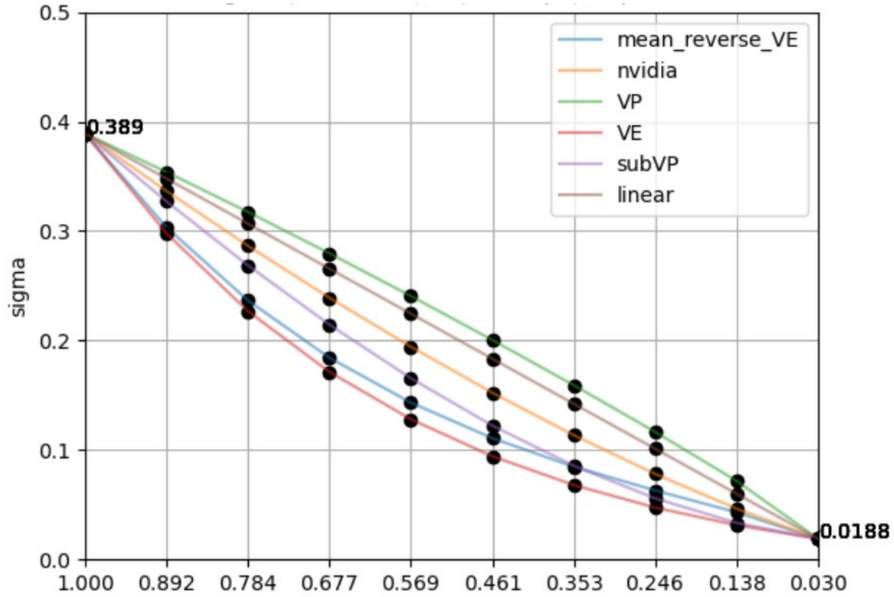


Methodology

- Four experiments set up
  - Interchanged scheduler functions
  - Baseline scheduler with modified derivative
  - Baseline scheduler with modified timesteps
  - Baseline scheduler with non-uniform timesteps
- Quantified through perceptual metrics
  - STOI (0-1), WARP-Q (.. – 0), DNSMOS and PESQ (1 – 5)
- Results generated with *max* value per each metric
- Tests performed with a pseudo-random 35 datapoint test set
- Fixed model, stochasticity, sampling routine

Experiment 1
Interchanged Scheduler Functions

- Baseline function *mean_reverse_VE* swapped with 5 other functions
- Given 0.5 – 0.05 range, normalize to effective values of the baseline



Experiment 1

Interchanged Scheduler Functions

		PESQ	STOI	WARPQ	DNSMOS
Linear	n=10	**2.147** (± 0.617)	**0.911** (± 0.06)	0.776 (± 0.199)	2.909 (±0.194)
	n=30	2.287 (± 0.631)	**0.923** (± 0.057)	0.755 (± 0.187)	2.934 (± 0.186)
SubVP	n=10	1.503 (± 0.261)	0.891 (± 0.062)	0.884 (± 0.142)	2.549 (± 0.252)
	n=30	2.202 (± 0.631)	**0.923** (± 0.057)	0.775 (± 0.193)	2.898 (± 0.197)
VE	n=10	1.62 (± 0.349)	0.885 (± 0.064)	0.907 (± 0.149)	2.619 (± 0.289)
	n=30	2.13 (± 0.641)	0.92 (± 0.058)	0.774 (± 0.187)	2.861 (± 0.208)
VP	n=10	2.122 (± 0.568)	0.907 (± 0.061)	**0.774** (± 0.192)	**2.925** (± 0.184)
	n=30	**2.31** (± 0.609)	0.92 (± 0.059)	**0.748** (± 0.179)	**2.96** (± 0.179)
Nvidia	n=10	1.883 (± 0.493)	0.908 (± 0.062)	0.812 (± 0.177)	2.824 (± 0.212)
	n=30	2.249 (± 0.64)	**0.923** (± 0.057)	0.759 (± 0.188)	2.909 (± 0.196)
Baseline	n=10	1.49 (± 0.249)	0.884 (± 0.067)	0.896 (± 0.139)	2.561 (± 0.265)
	n=30	2.175 (± 0.671)	0.922 (± 0.056)	0.765 (± 0.198)	2.927 (± 0.187)

NOISY

BASELINE

N=10

N=30

VP

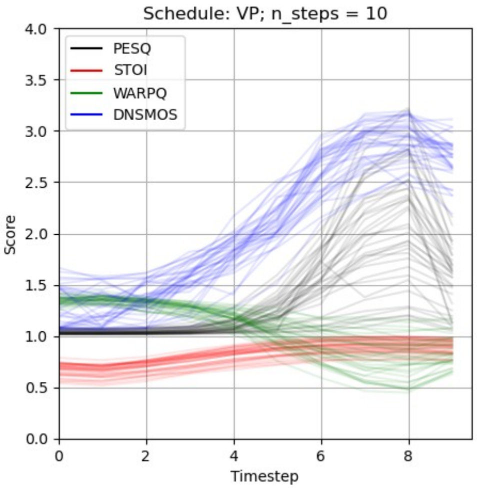
N=10

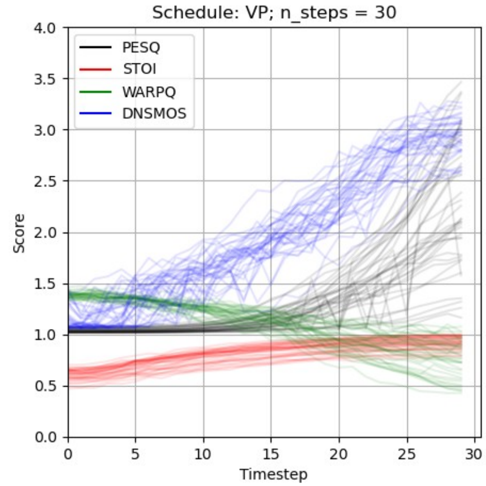
N=30

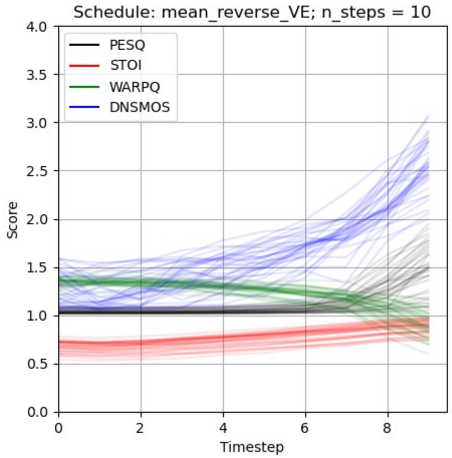
CLEAN

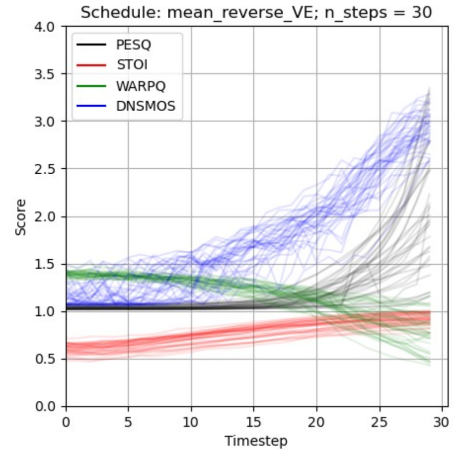
"Downing street will make the second appointment in the Scotland office today.."

Experiment 1
Interchanged Scheduler Functions





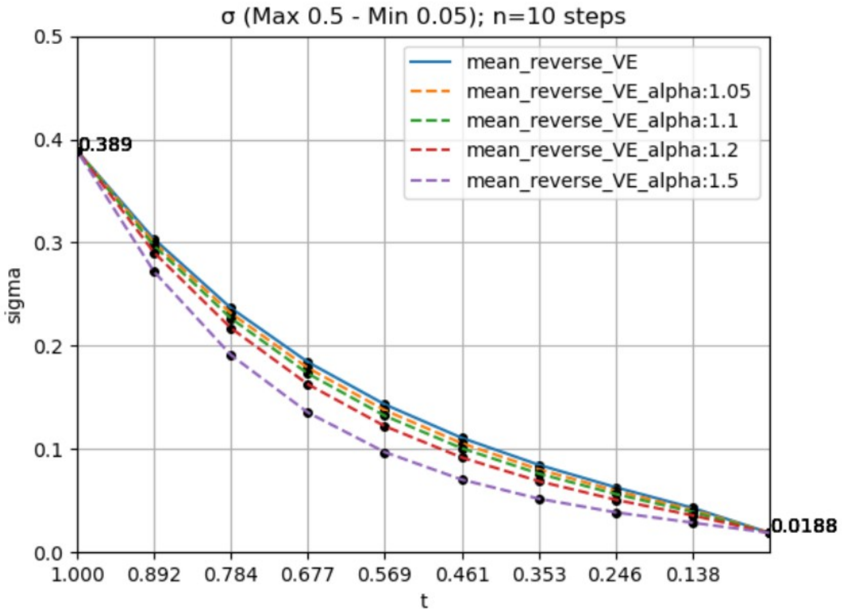




Experiment 2

Baseline Function with Modified Derivative

- Introduce an **α** modifier, vary the curve steepness
- Idea: significant improvement threshold decrease with potentially higher peak performance



Experiment 2

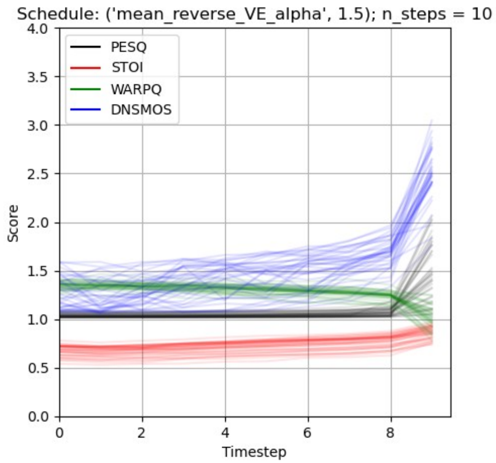
Baseline Function with Modified Derivative

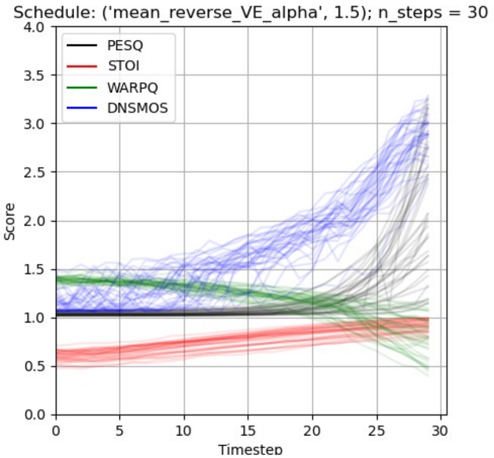
- N=10 mild-to-none increase in performance
- N=30 mild-to-none decrease of performance
- No clear winner for the modifications

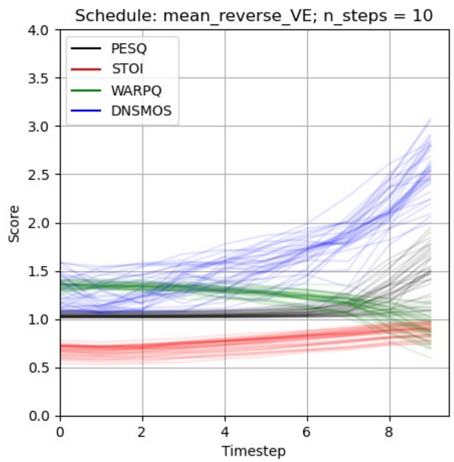
		PESQ	STOI	WARPQ	DNSMOS
Alpha=1.05	n=10	1.498 (± 0.26)	0.882 (± 0.067)	0.905 (± 0.139)	2.563 (± 0.272)
	n=30	2.17 (± 0.671)	0.921 (± 0.056)	**0.764** (± 0.2)	**2.928** (± 0.186)
Alpha=1.1	n=10	1.513 (± 0.272)	0.881 (± 0.067)	0.913 (± 0.14)	2.566 (± 0.28)
	n=30	2.167 (± 0.67)	0.921 (± 0.057)	0.768 (± 0.201)	2.912 (± 0.191)
Alpha=1.2	n=10	**1.548** (± 0.304)	0.879 (± 0.066)	0.928 (± 0.144)	**2.586** (± 0.287)
	n=30	2.156 (± 0.67)	0.921 (± 0.057)	0.767 (± 0.2)	2.912 (± 0.194)
Alpha=1.5	n=10	1.504 (± 0.296)	0.864 (± 0.064)	1.017 (± 0.114)	2.498 (± 0.284)
	n=30	2.089 (± 0.68)	0.916 (± 0.06)	0.782 (± 0.206)	2.888 (± 0.217)
Baseline	n=10	1.49 (± 0.249)	**0.884** (± 0.067)	**0.896** (± 0.139)	2.561 (± 0.265)
	n=30	**2.175** (± 0.671)	**0.922** (± 0.056)	0.765 (± 0.198)	2.927 (± 0.187)

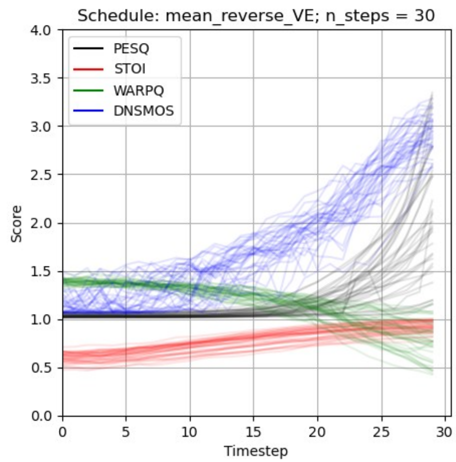
Experiment 2

Baseline Function with Modified Derivative





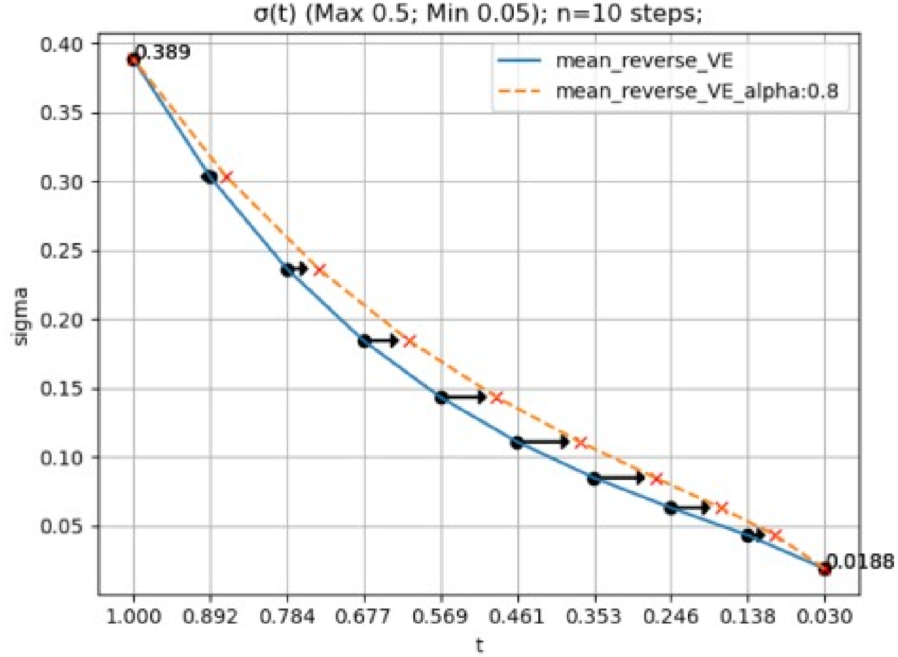




Experiment 3

Baseline Function with Timestep Offset

- Apply an **α** modifier and project the unmodified σ
- Idea: Compress the sampling points, decrease the amount of noise per step



Ernests Lavrinovics
MED10, AAU-CPH

Experiment 3

Baseline Function with Timestep Offset

- Consistent improvement over baseline
- Effect decays with increase of step size

		PESQ	STOI	WARPQ	DNSMOS
Alpha=0.8	n=10	**1.945** (± 0.526)	**0.903** (± 0.064)	**0.834** (± 0.169)	**2.809** (± 0.234)
	n=15	**2.135** (± 0.606)	**0.915** (± 0.06)	**0.779** (± 0.189)	**2.918** (± 0.227)
	n=30	**2.211** (± 0.669)	**0.923** (± 0.55)	**0.759** (± 0.201)	**2.944** (± 0.181)
Baseline	n=10	1.49 (± 0.249)	0.884 (± 0.067)	0.896 (± 0.139)	2.561 (± 0.265)
	n=15	1.93 (± 0.471)	0.911 (± 0.06)	0.805 (± 0.178)	2.858 (± 0.238)
	n=30	2.175 (± 0.671)	0.922 (± 0.56)	0.765 (± 0.198)	2.927 (± 0.187)

NOISY

BASELINE

N=10

N=15

Alpha 0.8

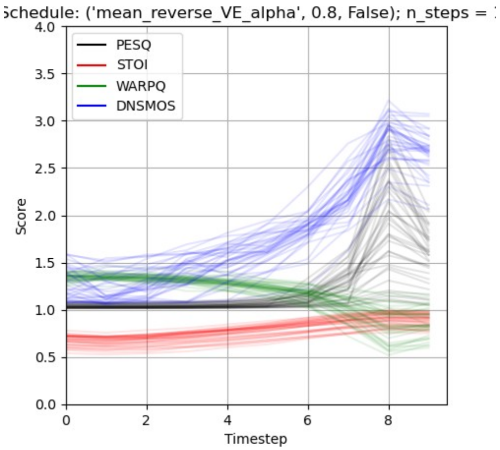
N=10

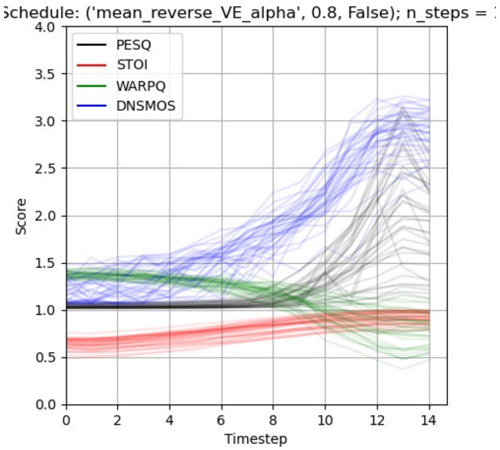
N=15

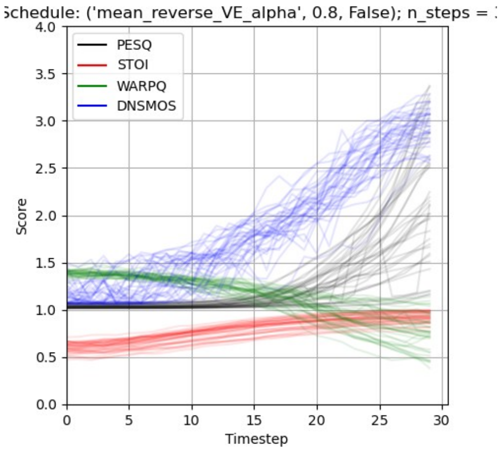
CLEAN

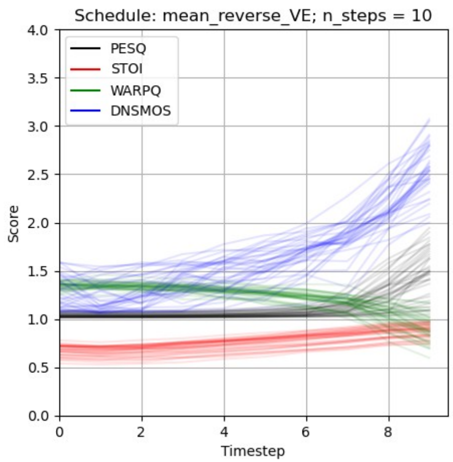
Experiment 3

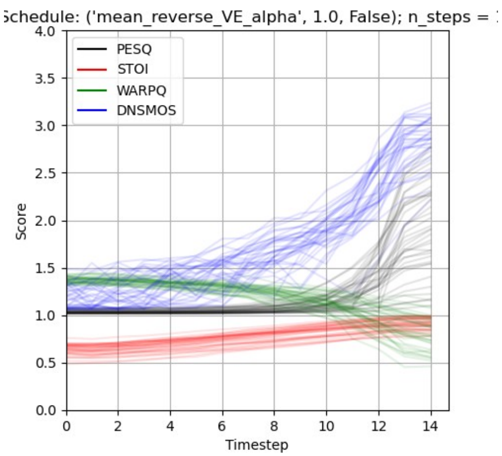
Baseline Function with Timestep Offset

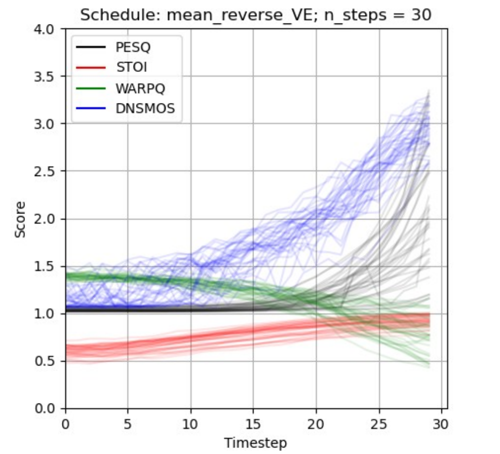








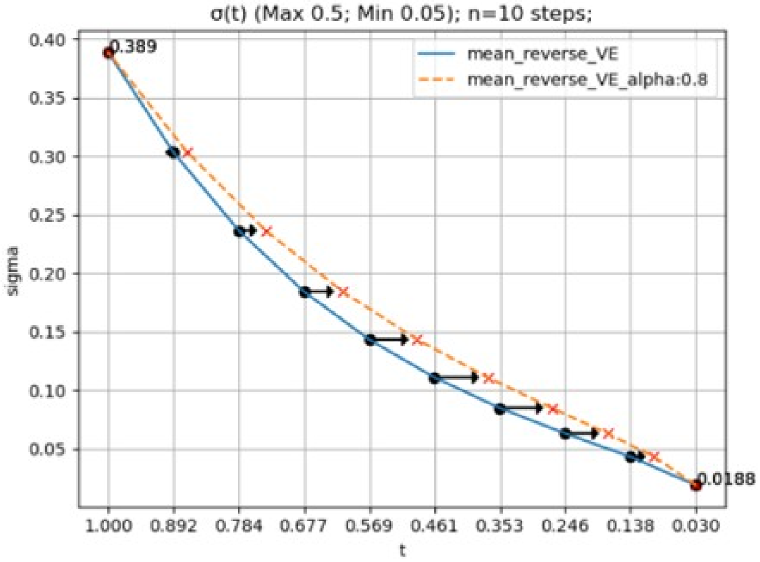


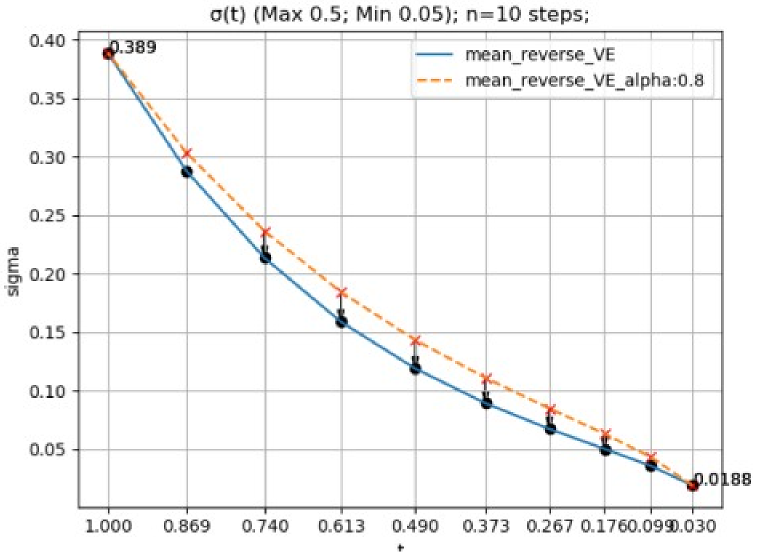


Experiment 4

Baseline Function with Non-uniform Timesteps

- Exp 3. schedule σ projected back to the baseline
- Idea: Compress sampling points towards the end, move back to familiar σ-t ratios





Experiment 4

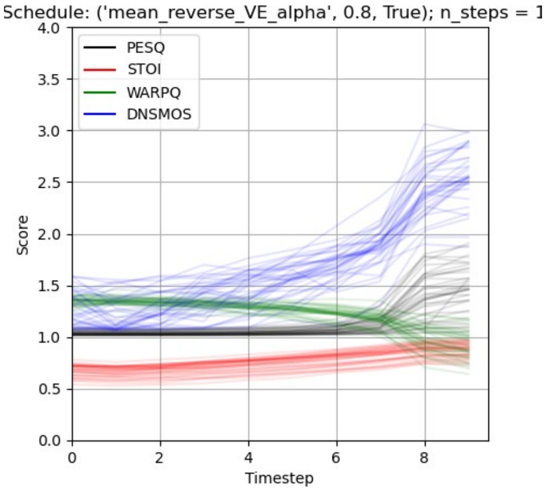
Baseline Function with Non-uniform Timesteps

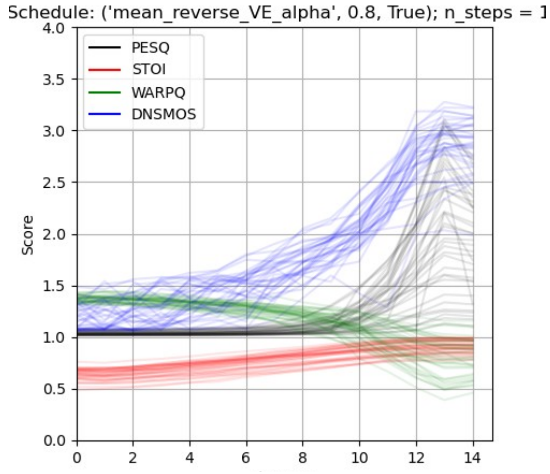
- Minor-to-none improvements in performance
- More performant means – higher std.dev.

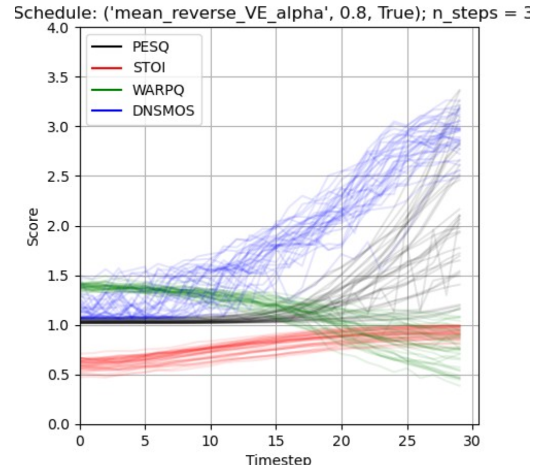
		PESQ	STOI	WARPQ	DNSMOS
	n=10	1.464 (± 0.238)	0.878 (± 0.067)	0.912 (± 0.134)	**2.567** (± 0.265)
Alpha=0.8	n=15	**2.1** (± 0.603)	0.91 (± 0.061)	**0.791** (± 0.193)	**2.884** (± 0.253)
	n=30	**2.187** (± 0.671)	0.922 (± 0.056)	**0.761** (± 0.203)	**2.936** (± 0.192)
Baseline	n=10	**1.49** (± 0.249)	**0.884** (± 0.067)	**0.896** (± 0.139)	2.561 (± 0.265)
	n=15	1.93 (± 0.471)	**0.911** (± 0.06)	0.805 (± 0.178)	2.858 (± 0.238)
	n=30	2.175 (± 0.671)	0.922 (± 0.56)	0.765 (± 0.198)	2.927 (± 0.187)

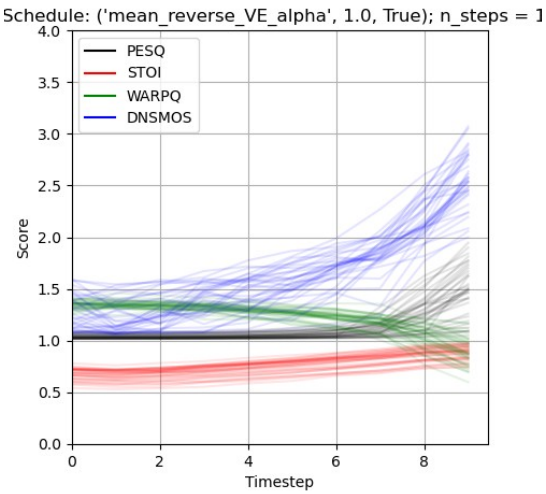
Experiment 4

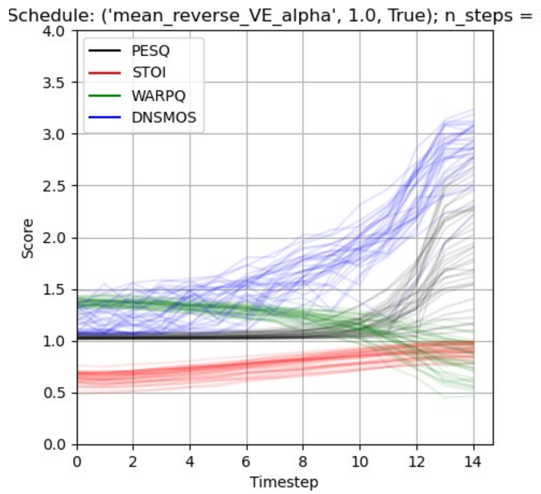
Baseline Function with Non-uniform Timesteps

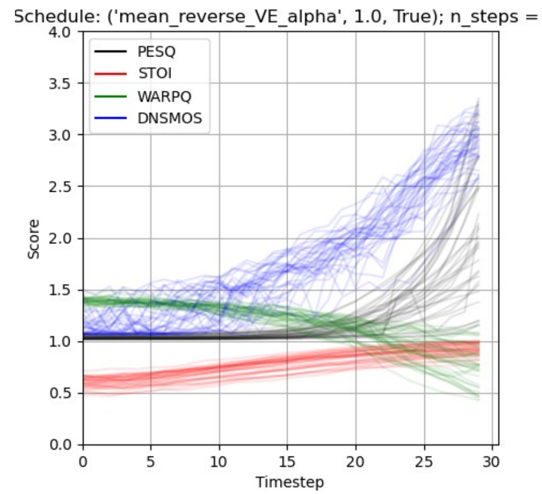












Summary

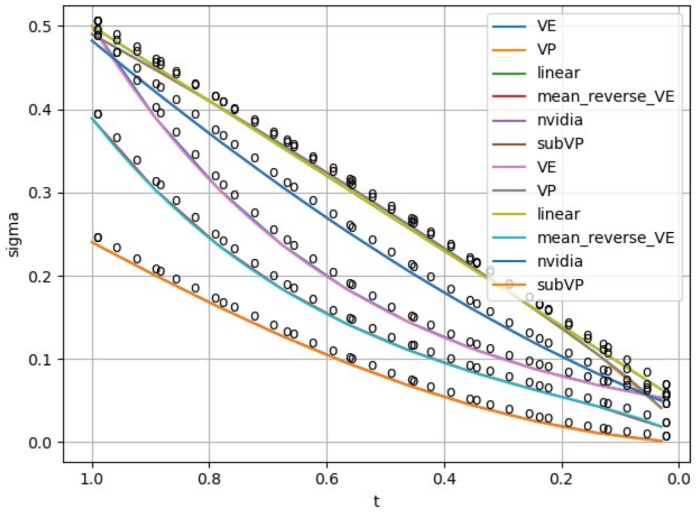
- Interchanged schedule and modified baseline may improve the baseline performance
  - Theory suggests that L2 models tend to remove too much noise
- Effects decline with increased timesteps
  - Robustness of more gradual diffusion
- Quality in may be more dependent on the progression rather than discretization itself
  - Exp 4 compression of timesteps alone does not yield improvements and may degrade results

Thank you!

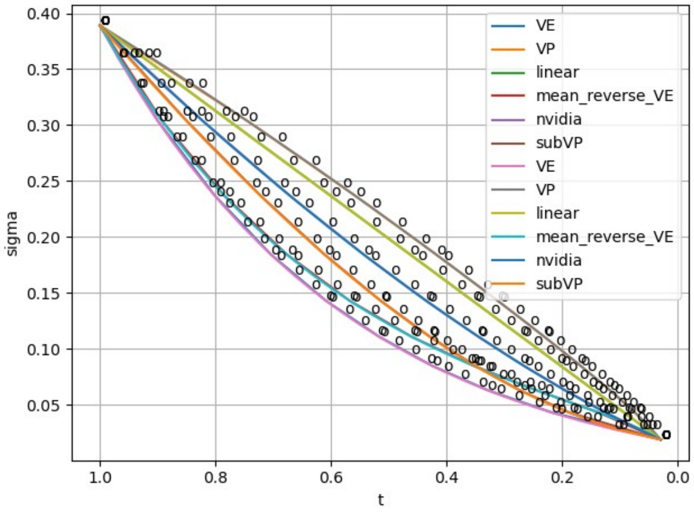
Backup 1

Experiment 1: Normalized vs Unnormalized functions

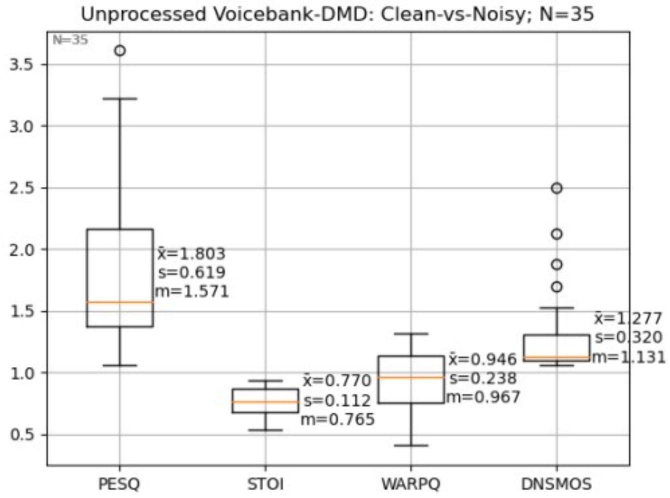
- For range (0.5 – 0.05) different effective values
- Normalize via linear scaling -> $f(\sigma, x, y) = \frac{(\sigma - \sigma_{min})}{(\sigma_{max} - \sigma_{min})} * (y - x) + x$



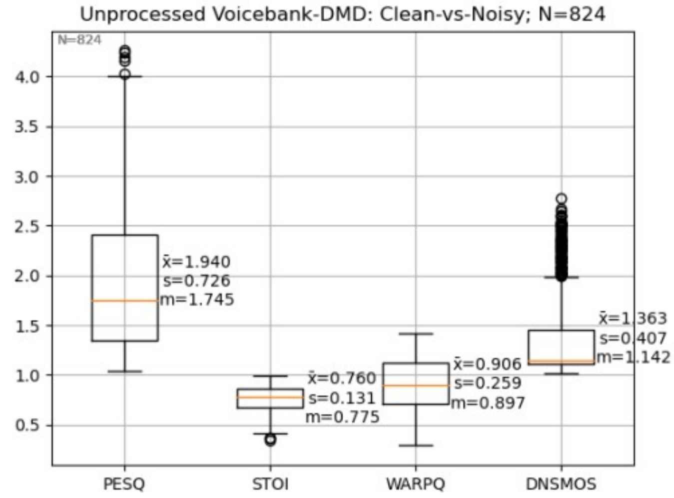
After normalization



Backup 2
Metrics of Unprocessed Dataset



(a) 35 random sample subsplit.



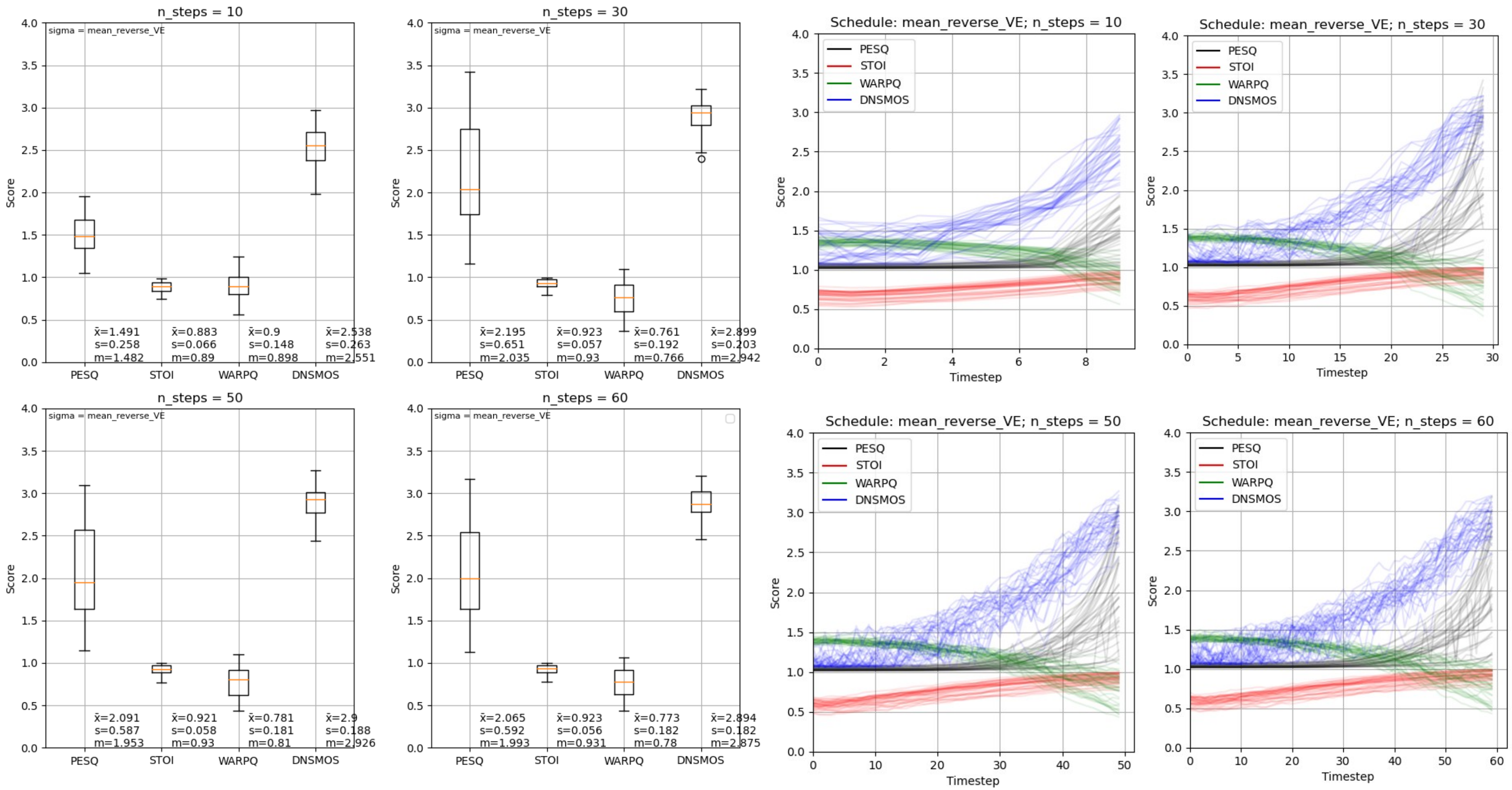
(b) Full dataset.

Baseline	n=10	1.49 (± 0.249)	0.884 (± 0.067)	0.896 (± 0.139)	2.561 (± 0.265)
	n=30	2.175 (± 0.671)	0.922 (± 0.056)	0.765 (± 0.198)	2.927 (± 0.187)

Baseline model processing results

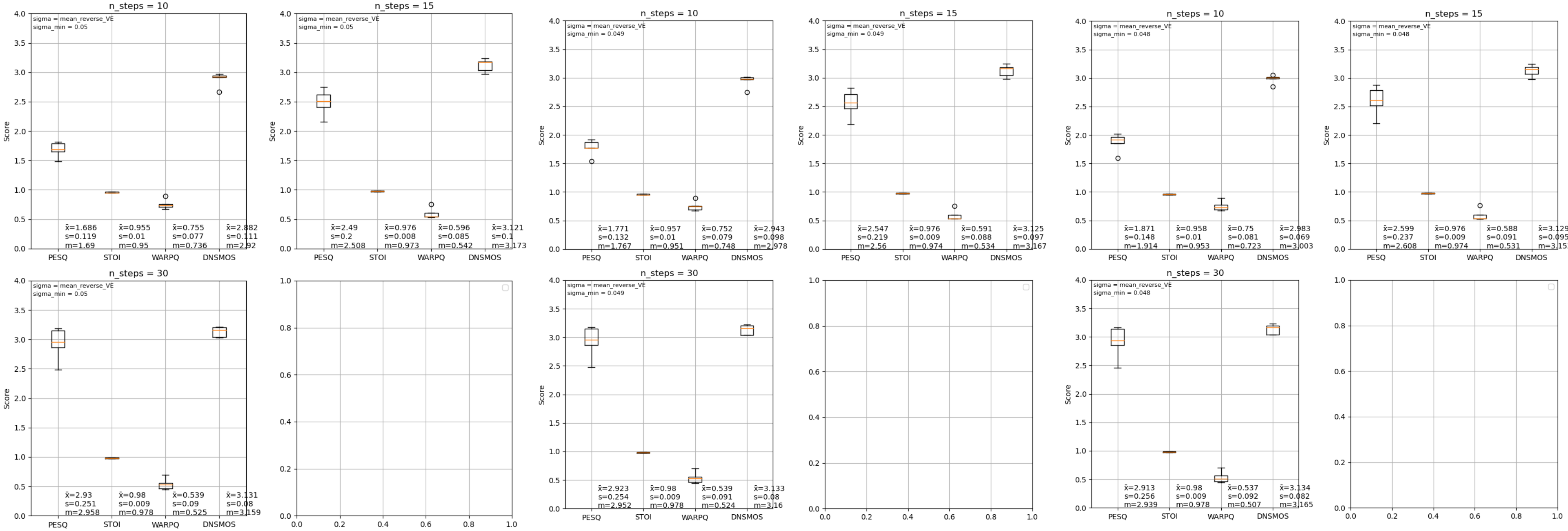
Backup 3

Further Increased Timesteps



Backup 4

Decreased Sigmas



Backup 4
Decreased Sigmas