# Speech Signal Processing and Speech Enhancement Summer 2023

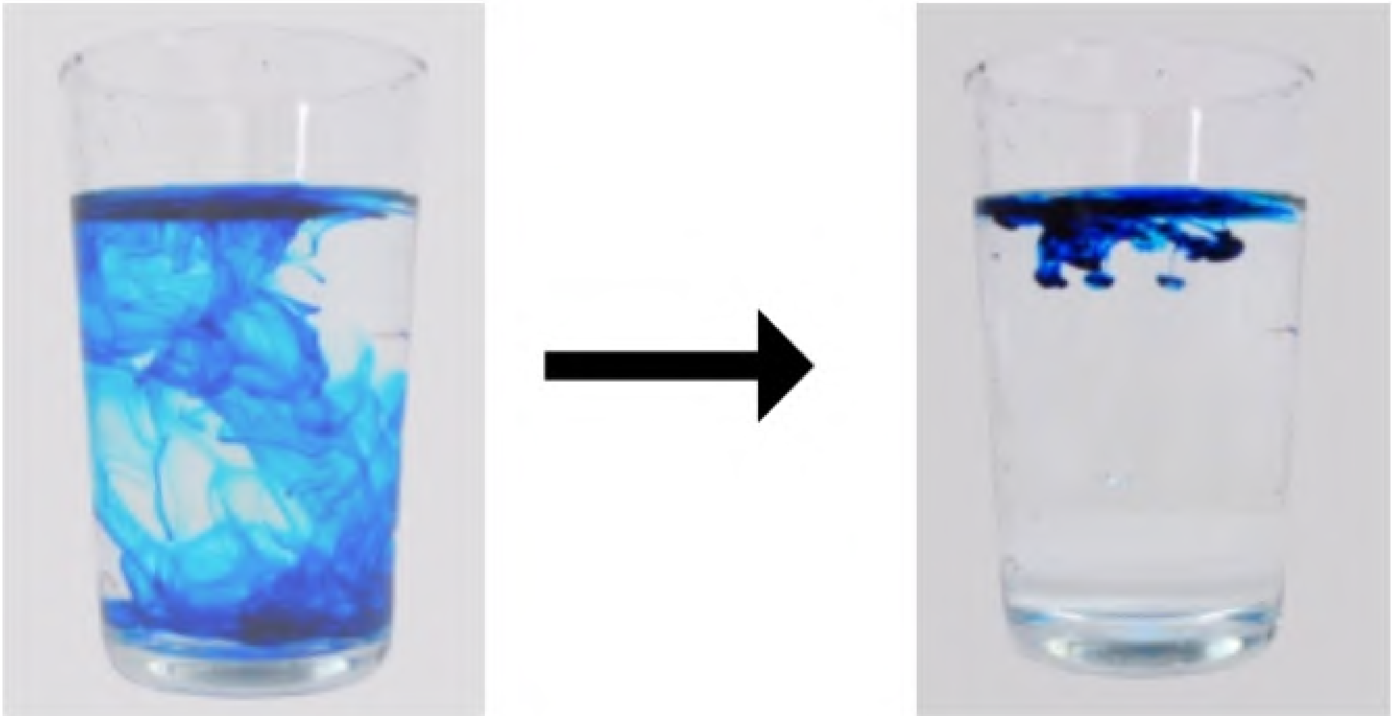
## Denoising Diffusion on Speaker Embeddings

Akshat Gupta

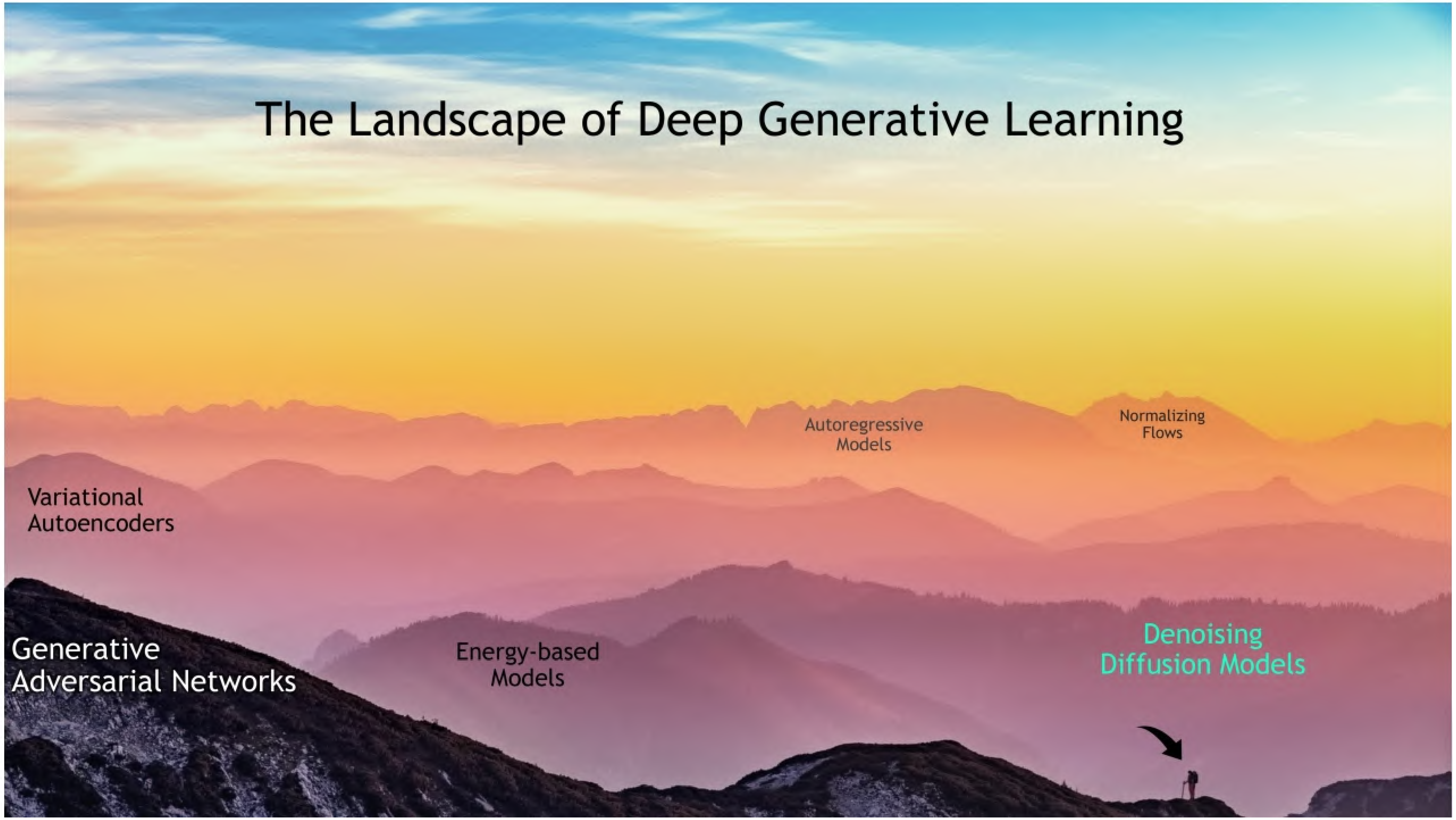
# Introduction

- Generative Deep Learning Model
- Eg. Audio Generation, Image Generation
- Uses Markov Chain to learn the distribution of data- mean, variance

Diffusion

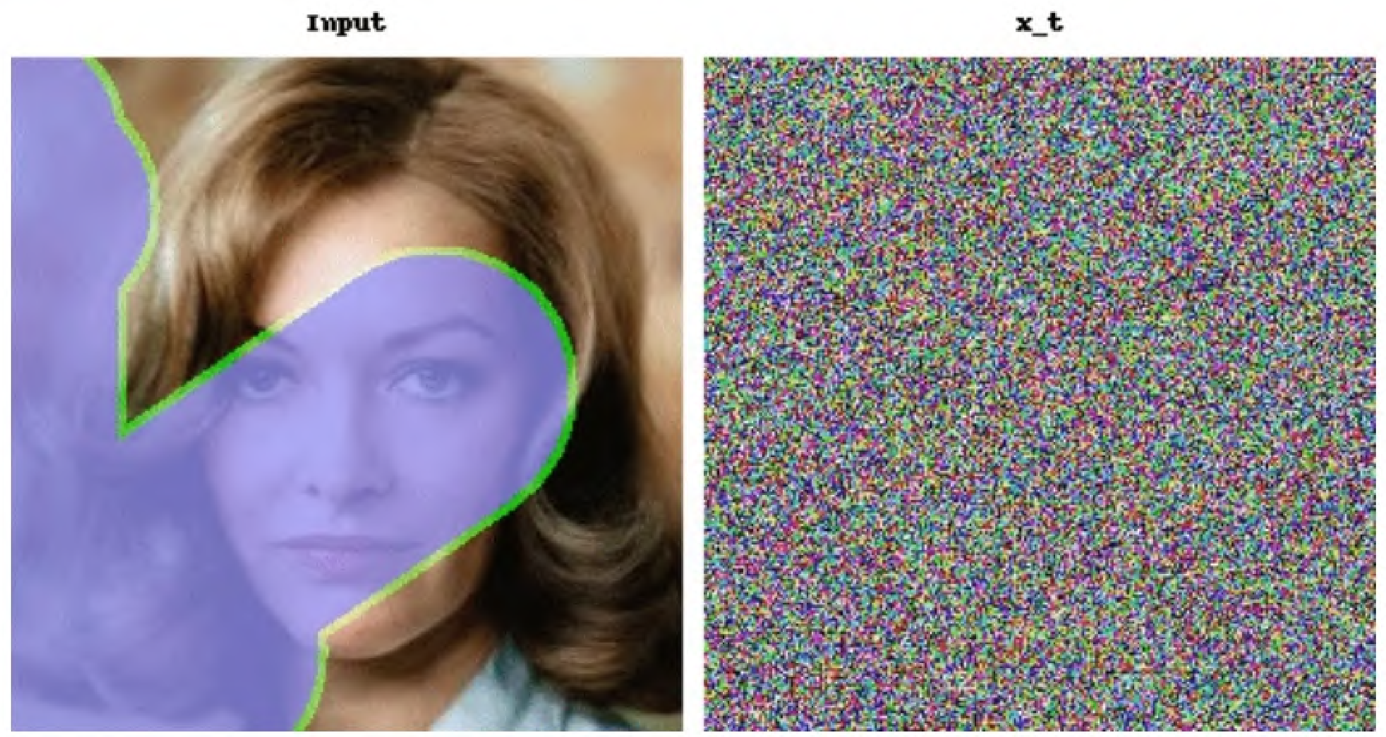


# Motivation


The Landscape of Deep Generative Learning
Autoregressive Models
Normalizing Flows
Variational Autoencoders
Generative Adversarial Networks
Energy-based Models
Denoising Diffusion Models

# Motivation

- Image Inpainting- Repaint
- Anomaly Detection - AnoDDPM




RePaint

Try Pitch

# Motivation

- Natural Language Processing- Text Synthesis
- Multi-Modal Learning- Text to Image -  Imagen, DALL-E, Text to Audio, Diff-TTS

A group of teady bears in suit in corporate
office celebrating the birthday

# Diffusion Model


Try Pitch

# Popularity of Diffusion Model

## Jonathan Ho

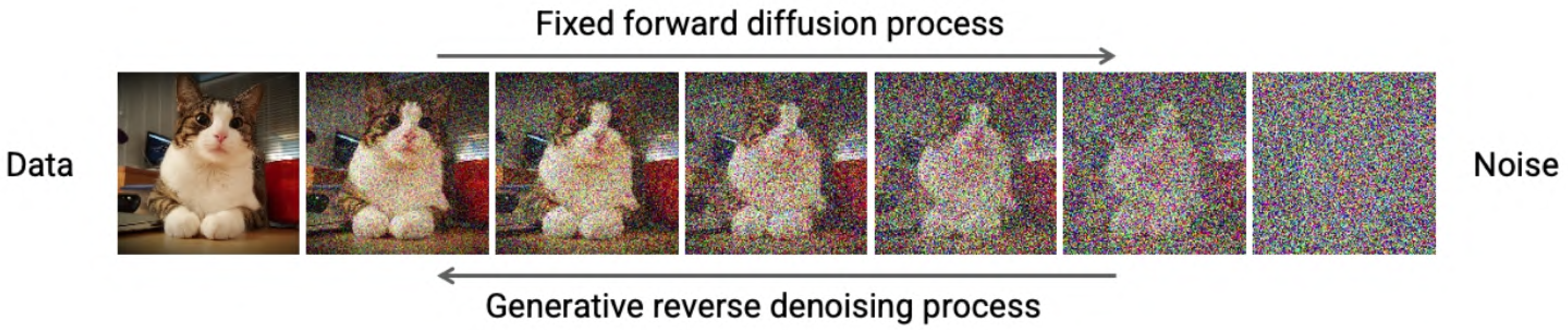
Unknown affiliation
Verified email at berkeley.edu - Homepage
Artificial Intelligence    Machine Learning

FOLLOW

| TITLE | CITED BY | YEAR |
|---|---|---|
| Generative adversarial imitation learning<br>J Ho, S Ermon<br>Advances in Neural Information Processing Systems, 4565-4573 | 2273 | 2016 |
| Evolution strategies as a scalable alternative to reinforcement learning<br>T Salimans, J Ho, X Chen, S Sidor, I Sutskever<br>arXiv preprint arXiv:1703.03864 | 1314 | 2017 |
| Denoising diffusion probabilistic models<br>J Ho, A Jain, P Abbeel<br>Advances in Neural Information Processing Systems 33, 6840-6851 | 1032 | 2020 |

# How DDPM Works?

- Learning to generate using denoising
- Two processes:
  - Forward Diffuson Process: adds noise to image
  - Reverse Denoising Process: generate new data



# Forward Diffusion Process

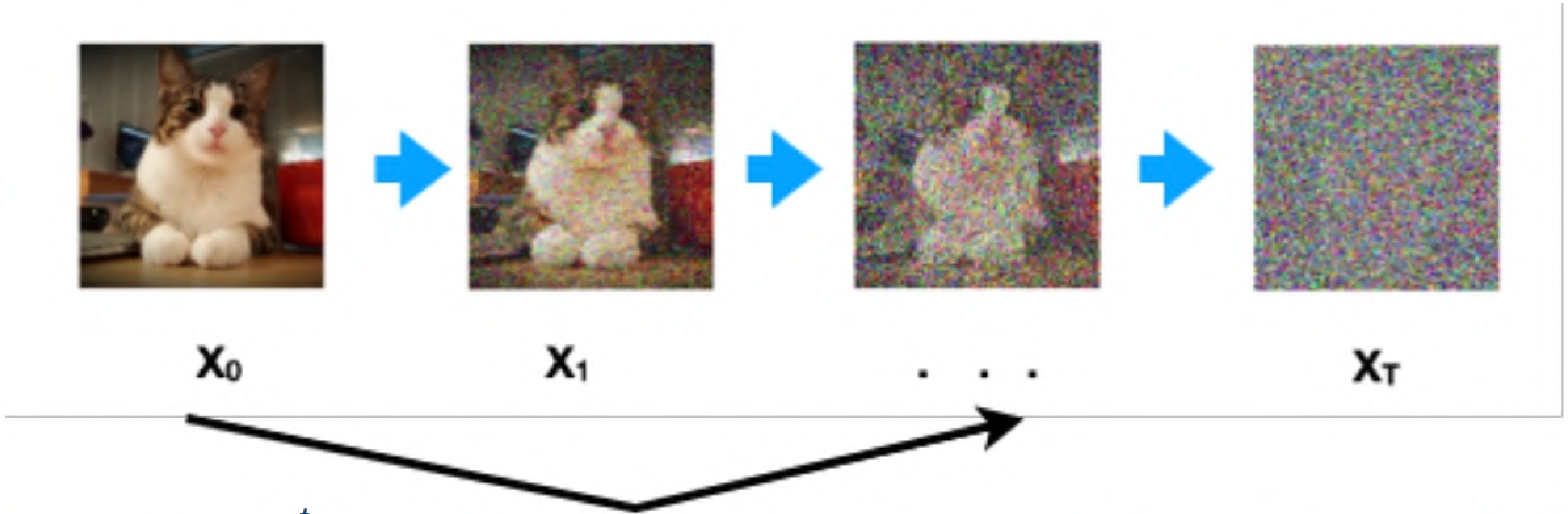
- Forward Process in T steps


Source: https://cvpr2022-tutorial-diffusion-models.github.io/

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) \longrightarrow q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \qquad (1)$$

- $\beta_t$ : variance at timestamp $t$, $\sqrt{1-\beta_t}$ : mean at timestamp $t$
- $q(x_t|x_{t-1})$ : distribution at each timestamp $t$, $q(x_{1:T}|x_0)$ : joint distribution

# Diffusion Kernel




$$\textit{Define} \quad \bar{\alpha}_t = \prod_{s=1}^{t} (1 - \beta_s) \longrightarrow q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}\right) \tag{2}$$

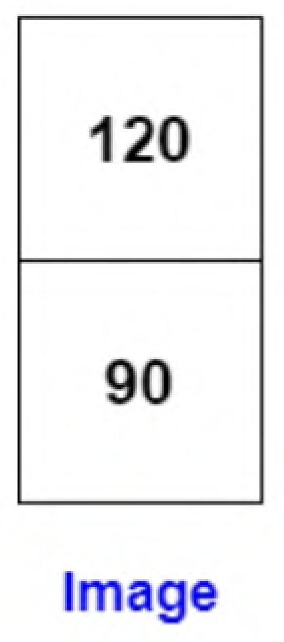
For sampling (reparameterization trick)

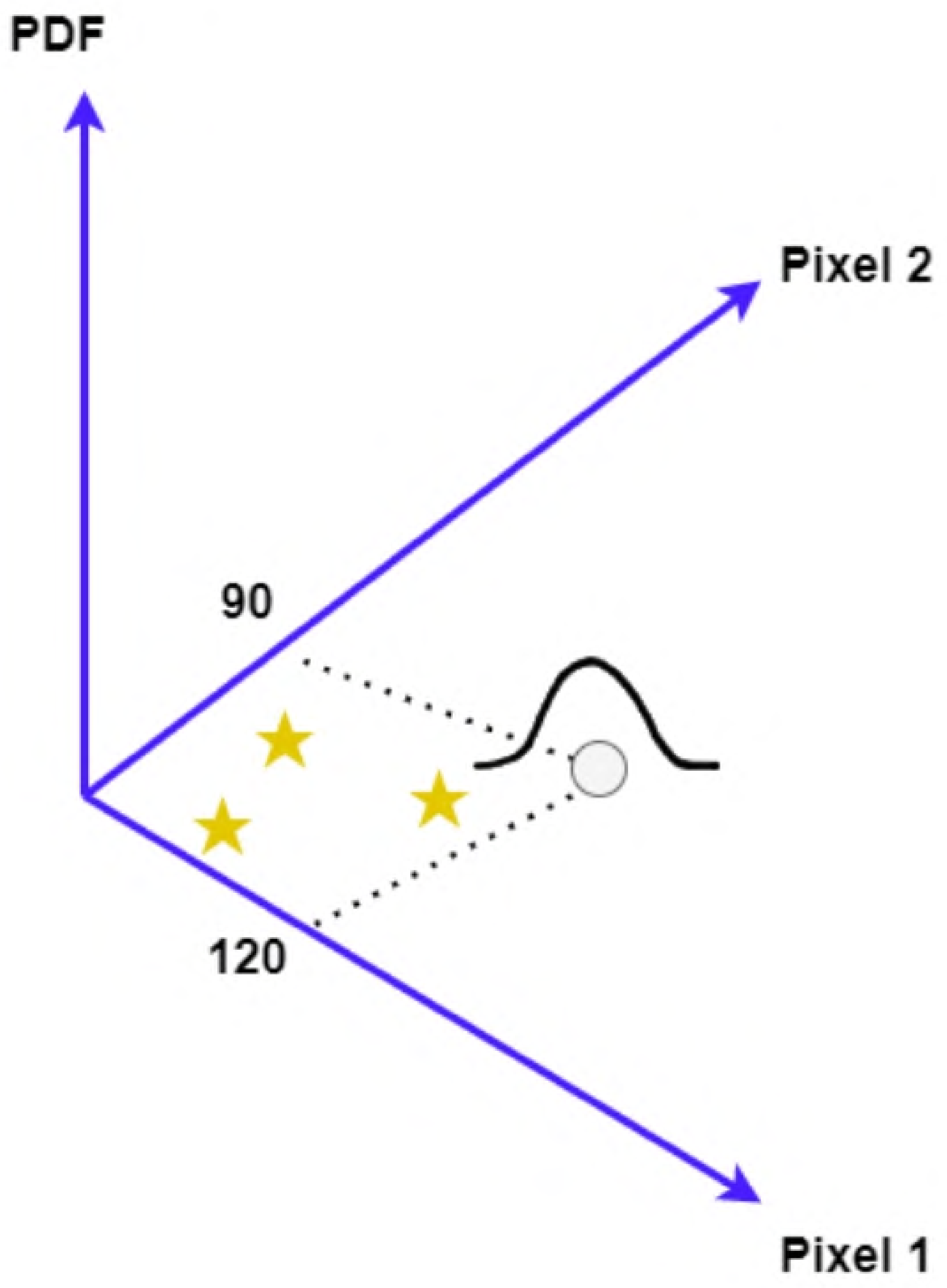
$$\mathbf{x_t} = \sqrt{\bar{\alpha}_t}\mathbf{x_0} + \sqrt{(1 - \bar{\alpha}_t)}\varepsilon \quad \textit{where} \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{3}$$

$\beta_t$ values scheduled (variance or noise schedule)

$$\bar{\alpha}_T \to 0 \textit{and}\ q(\mathbf{x}_T \mid \mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})) \tag{4}$$

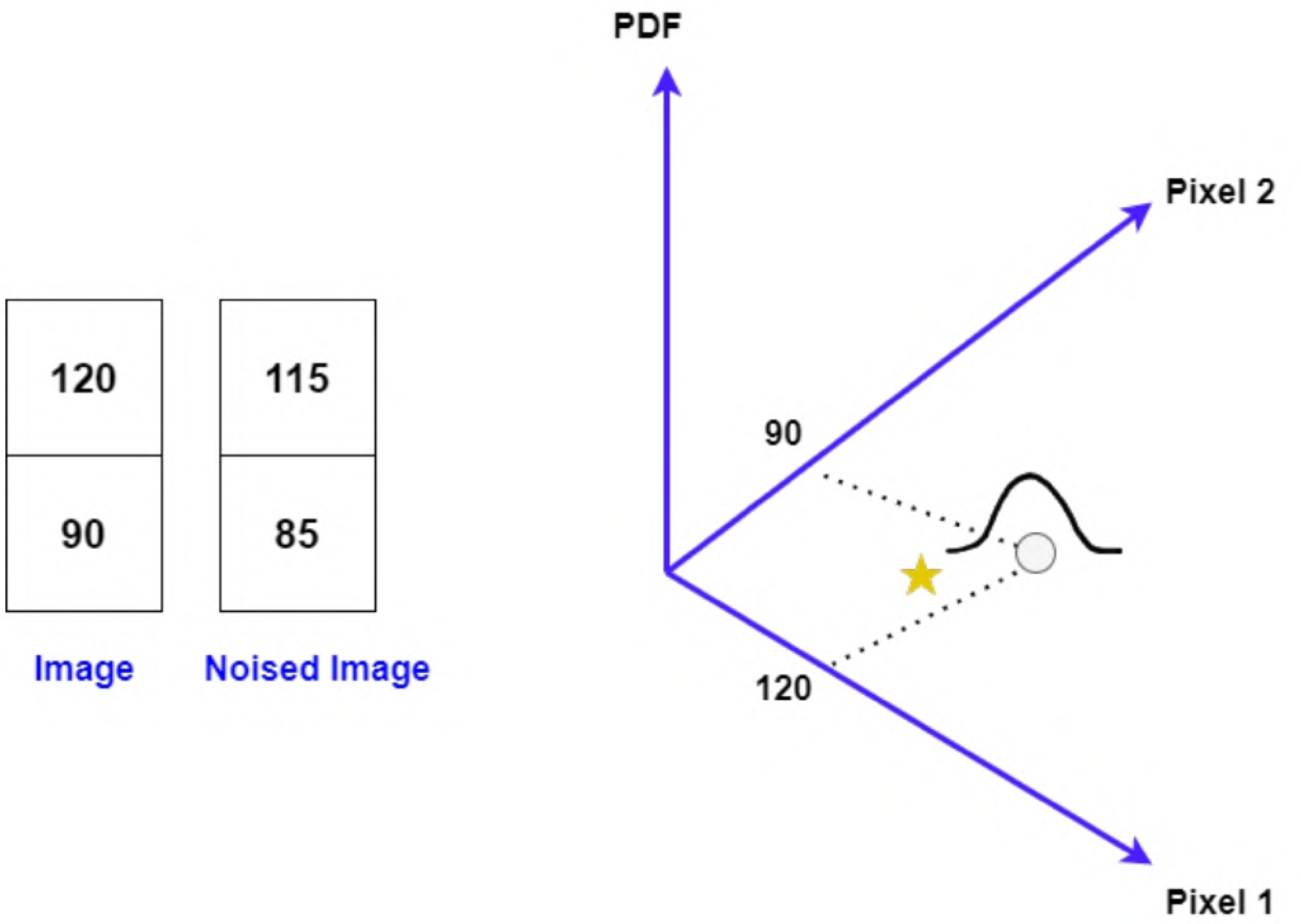
# How Diffusion happens?




Try Pitch

# How Diffusion happens?



# Reverse Diffusion Process

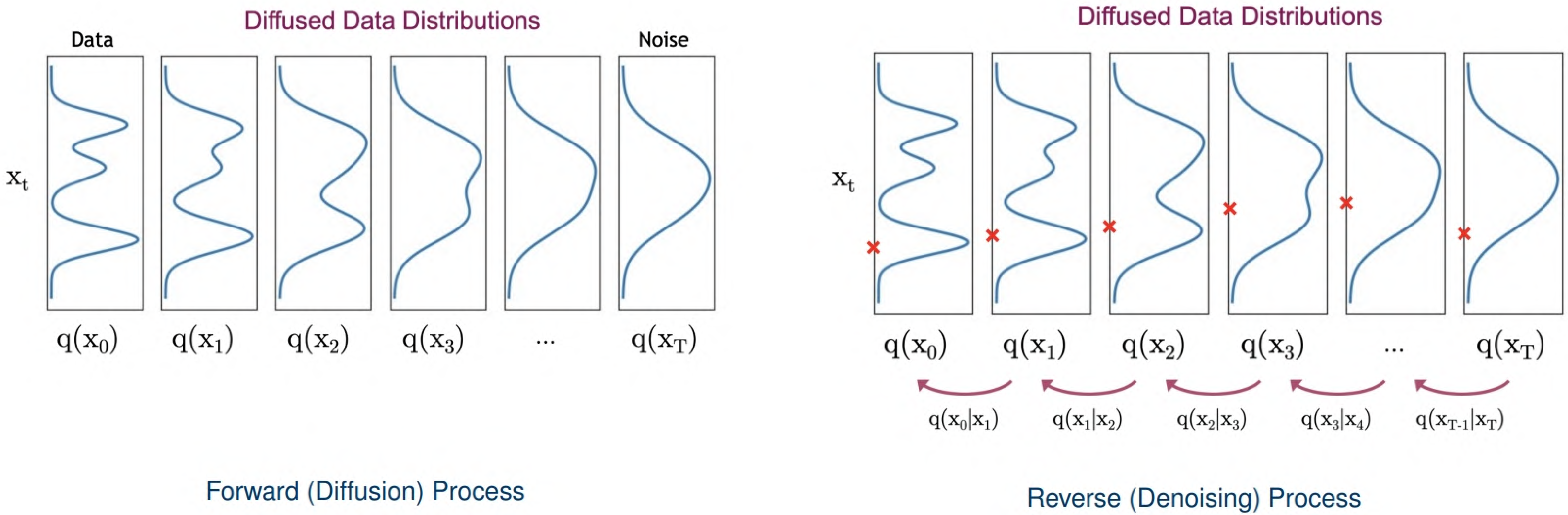

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}\right)$$

Trainable Network
(U-net, Denoising Autoencoder)

- $p_\theta(x_{t-1}|x_t)$ : denoising diffusion probability

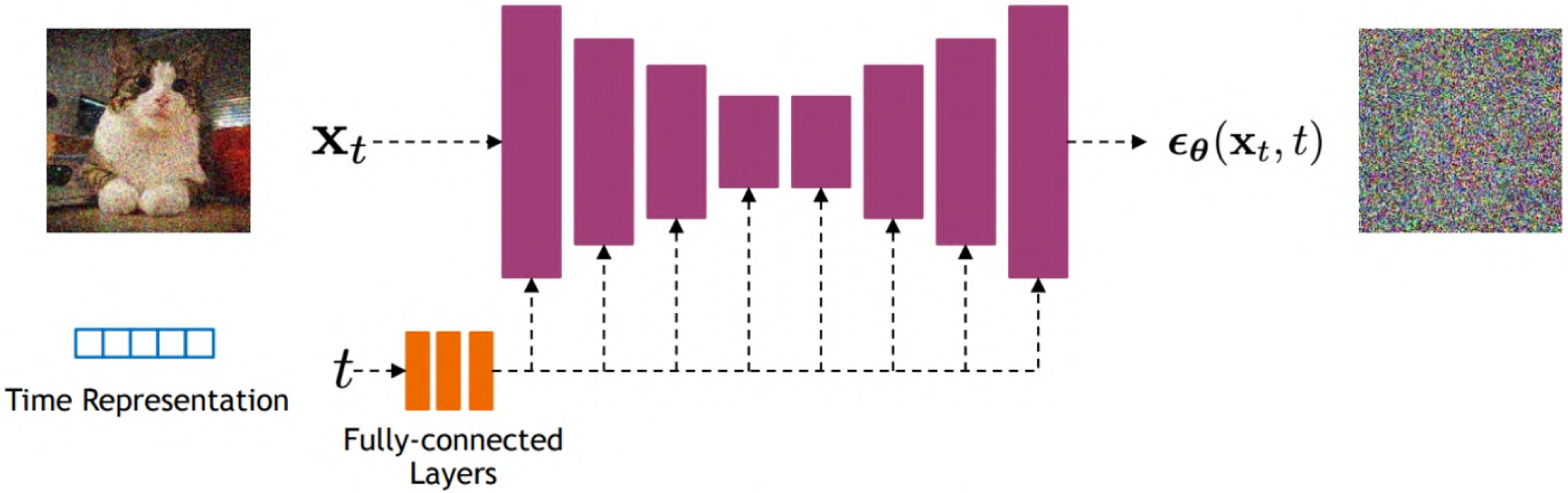
# What happens to distribution?




Source: `https://cvpr2022-tutorial-diffusion-models.github.io/`

# Network architecture

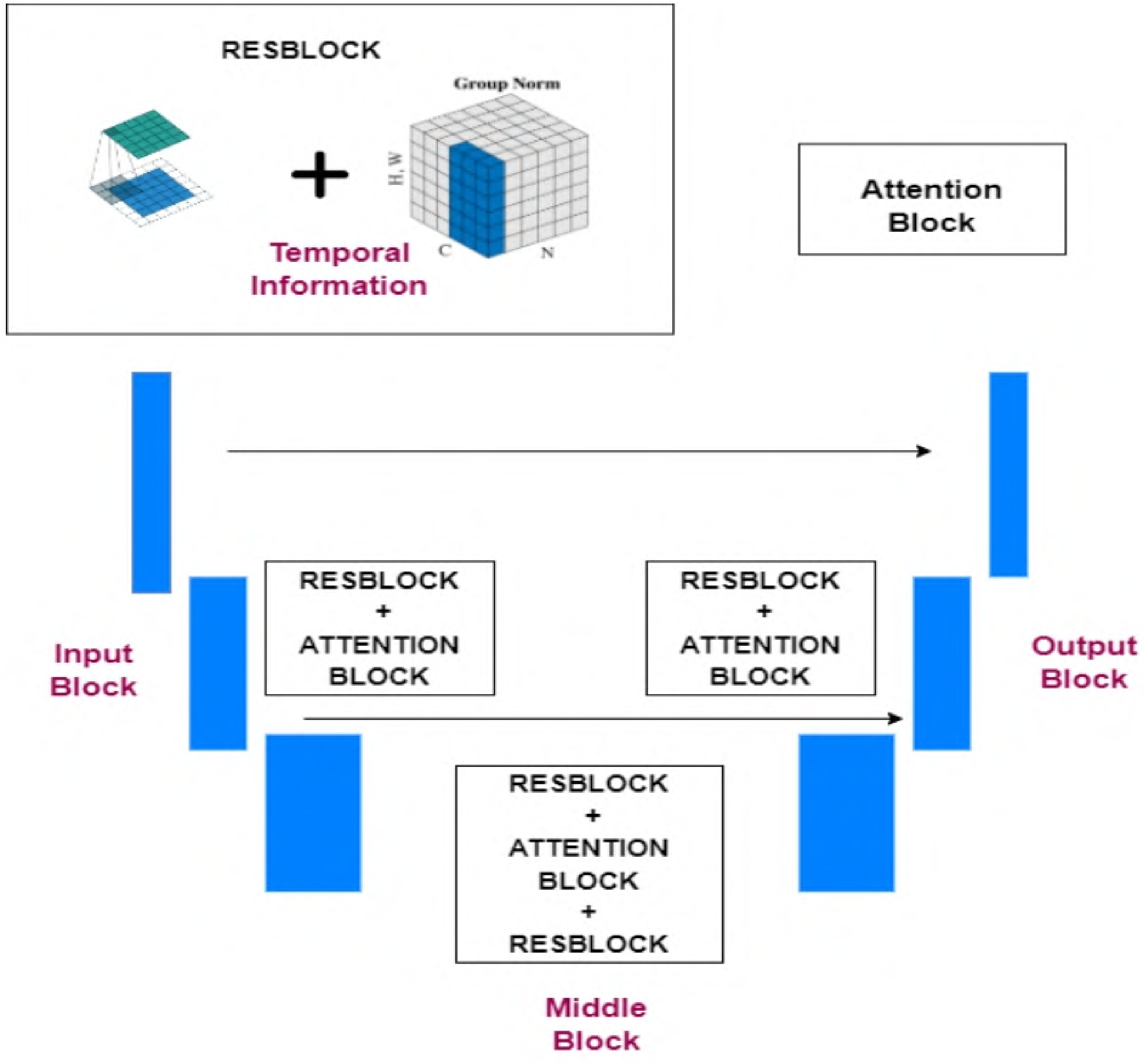
- U-NET architecture
- Train model predicts e_theta (noise)




$$L_{\mathbf{simple}} := \left\| \varepsilon - \varepsilon_\theta \left( \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t \right) \right\|^2$$

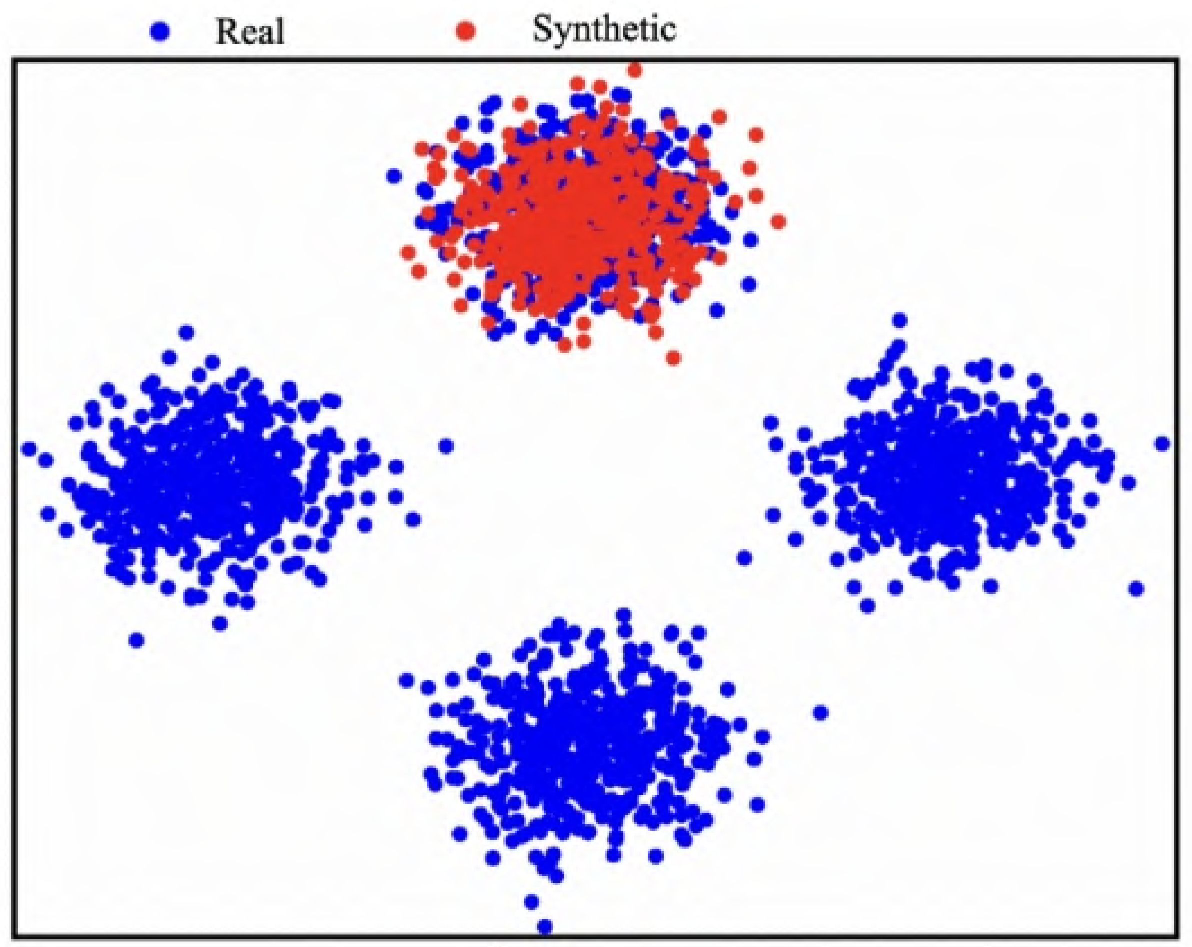
Source: https://cvpr2022-tutorial-diffusion-models.github.io/

# U-NET architecture



# Evaluation of DDPM

- Sample qualitative metrics
  - **Fidelity** - how realistic generated image is
  - **Diversity** - fake samples capture the variation
- Sample Quantitative metrics
  - **Inception Score(IS)**: sharpness and diversity, synthetic image != real images
  - **Frechet Inception Distance(FID)**: Distribution of synthetic images and real images
- **Mean and Variance** of the learned distributions and compare it to the real distribution



# Introduction to SpeakEMB

- Denoising Diffusion Probabilistic Models (DDPM)have emerged as prominent generative models.
- These models have been implemented predominantly on image synthesis
- Gradually moving to the domain of speech
- In this project, I aspire to use DDPM to generate Speaker Embedding.

# DDPM on Speaker Embeddings

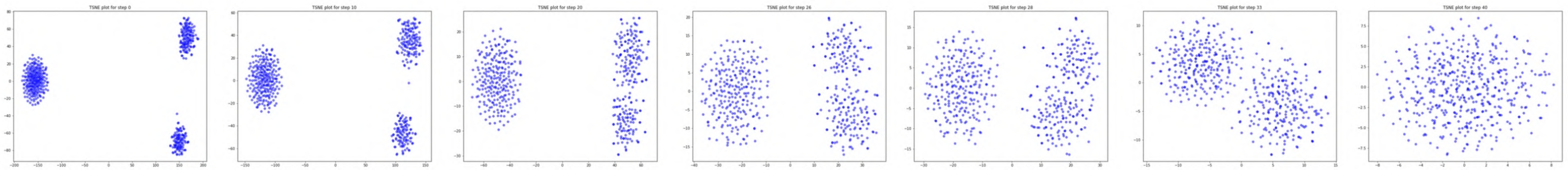
- Denoising Diffusion Model uses diffusion to add Gaussian Noise at every time step to the Speaker Embedding
- Denoising to remove the Gaussian Noise and generate a new set of Speaker Embedding

Forward Process: Diffusion

$$q(x_0) \to q(x_T) = \mathcal{N}(x_T, 0, I)$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t, x_{t-1}\sqrt{1-\beta_t}, I\beta_t)$$

The clean speaker embedding is converted into an isotropic Gaussian distribution in a step-by-step diffusion process.



# DDPM on Speaker Embeddings

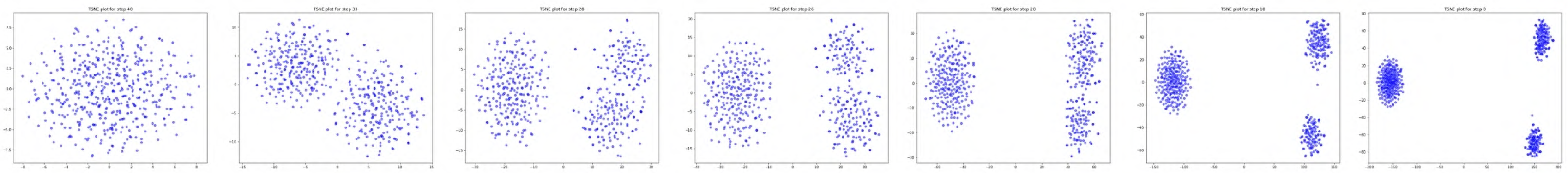
- When denoising is done on an isotropic Gaussian distribution, a new embedding can be generated

Backward Process: Denoising

$$p(x_0) = q(x_0) \leftarrow p(x_T) = \mathcal{N}(x_T; 0; I)$$

$$p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; f_\mu(x_t, t); f_{\sum}(x_t, t))$$

In the reverse process, it tries to gradually restore the clean input by predicting and removing the noise introduced in each step of the diffusion process.



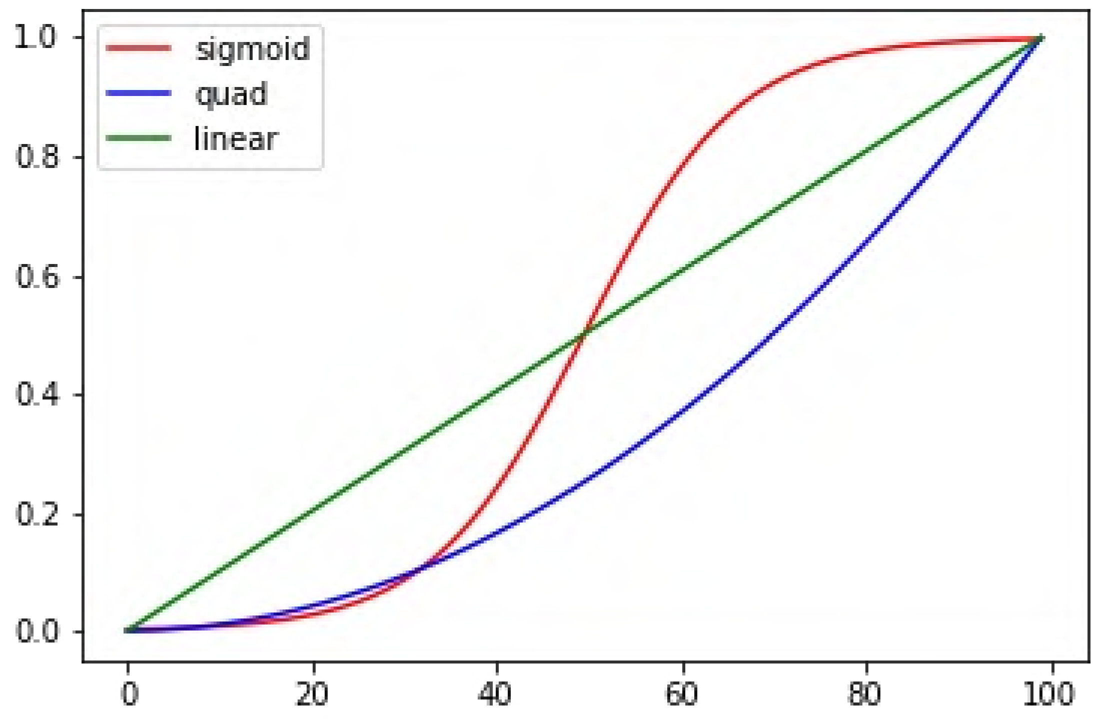

Try Pitch

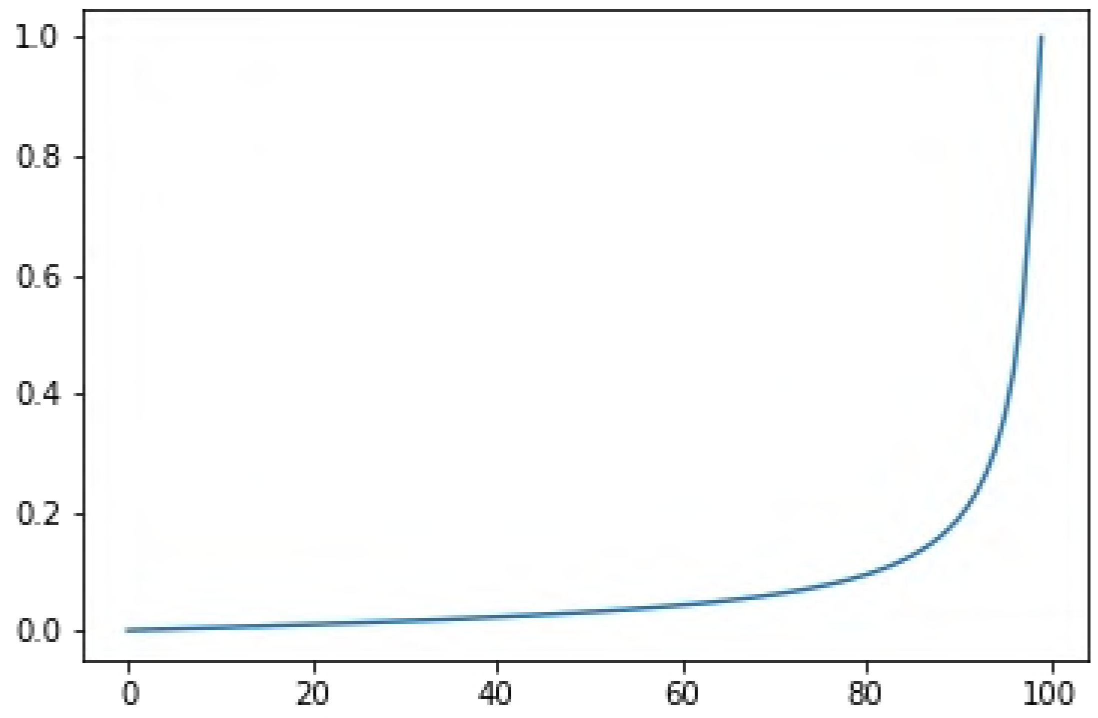
# Methods

- The baseline model is implemented on Denoising Diffusion Probabilistic Model to Generate Speaker Embedding
- Integrating the model architecture to a three layer fully connected Neural Network. Mount L1 loss in the baseline model and the following section.three-layer

# Variance Scheduling

- Four different Variance Schedulers of Linear, Sigmoid and Quadratic bearing ranges from e−5 to e−2
- Cosine bearing ranges from 0 to 1 on our Baseline Model with L1 loss
- The improvements in Cosine Variance Schedulers have shown promising results by Nichol et al.



# Dataset

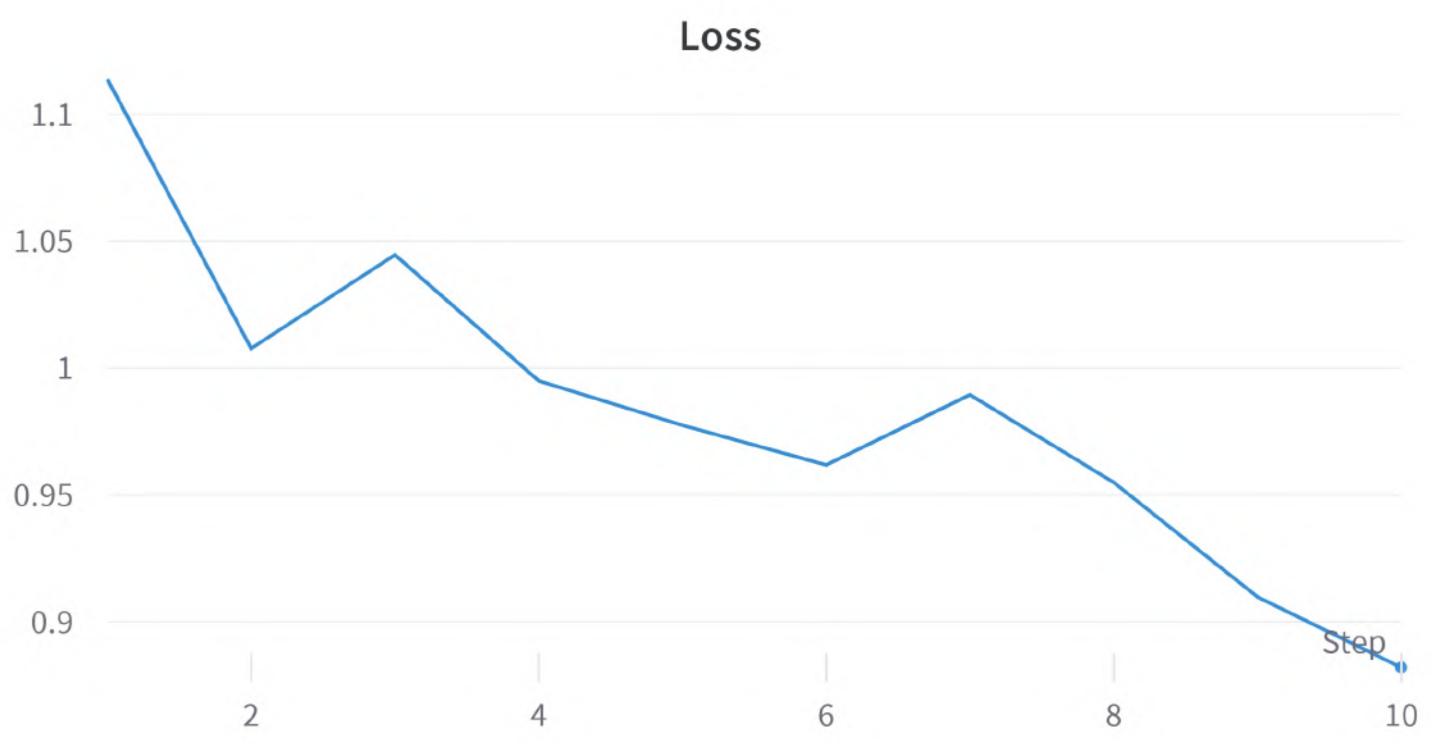
- Extracted speaker embedding on LibriSpeech dataset with Speechbrain Framework with ECAPA-TDNN Model, Hugging Face
- The extracted speaker embedding has different dimensionalities of 64, 128 and 704

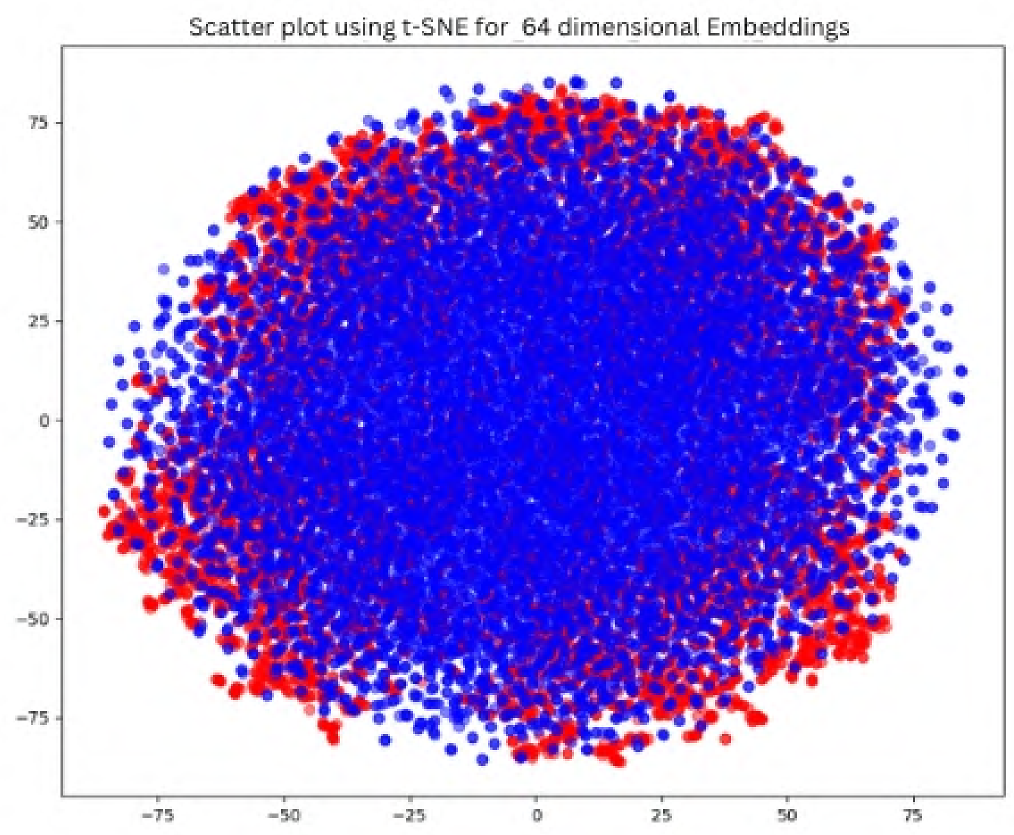
# Experiments

- The experiments are performed on three different sets of embedding having dimensions 64, 128 and 704
- The TSNE plot consists of the red data points which are corresponding to the original set of embedding and the blue data points which are corresponding to the generated set of embedding.
- The loss graph indicates learning and the convergence of the model.
- To visualize, we perform dimensionality reduction on the embedding and monitor the formation of clusters.

# 64 Dimension Experiment

- The fundamental experiment was performed on 64 dimension embedding until the generated embedding could
- be mapped into some clusters.
- The formation of these clusters is an illustrative explanation of how well the generation of the speaker embedding has been executed.

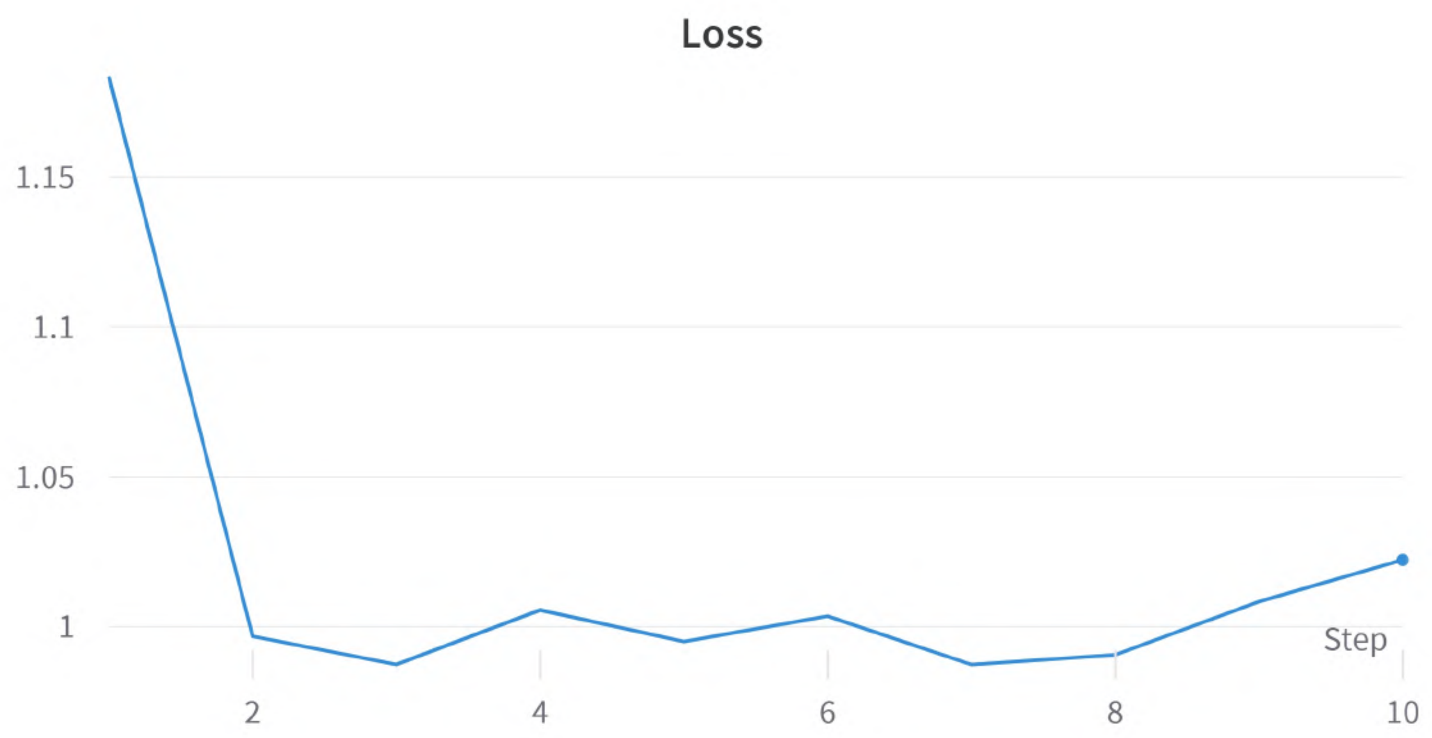


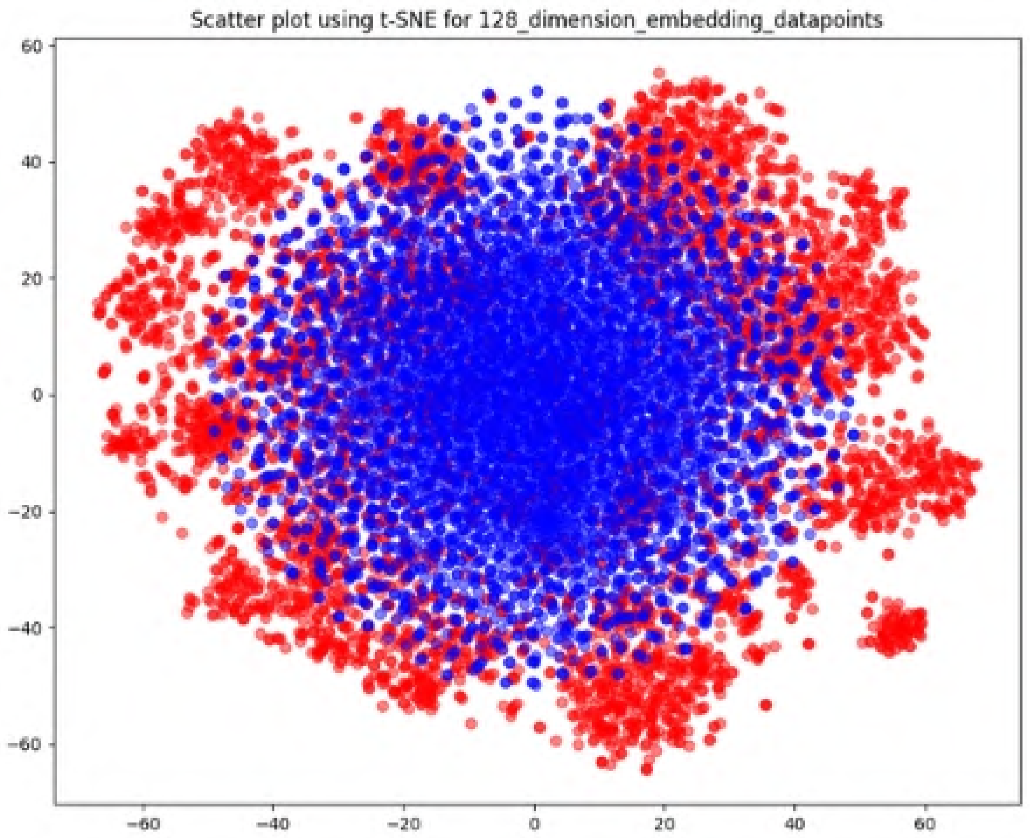




Try Pitch

# 128 Dimension Experiment

- With the increase in the number of the dimension of the speaker embedding from 64 to 128, the cluster formation of the generated embedding has been prominent.
- The blue clusters are constrained in the centre which shows that there has been a significant amount of learning but it definitely requires a large training cycle to fairly replicate the distribution of the original embedding.



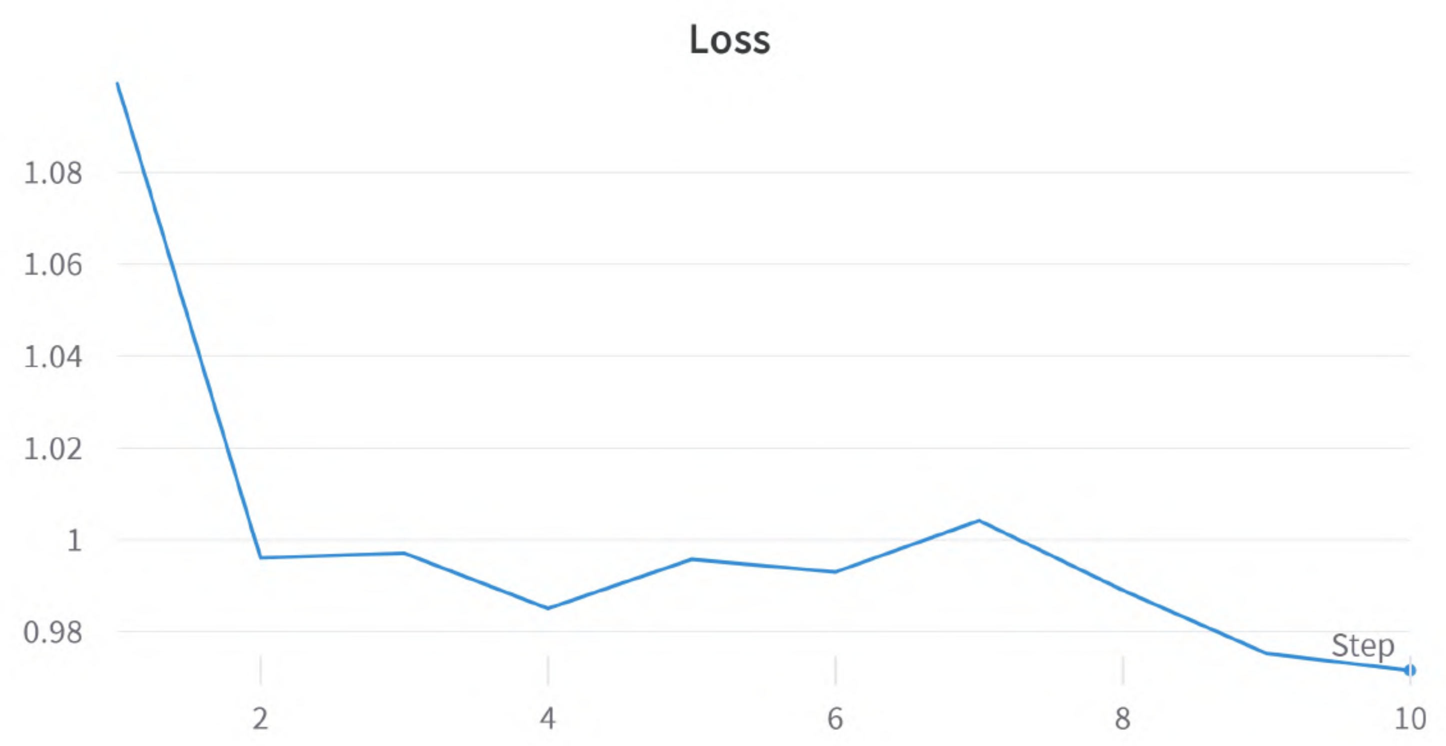



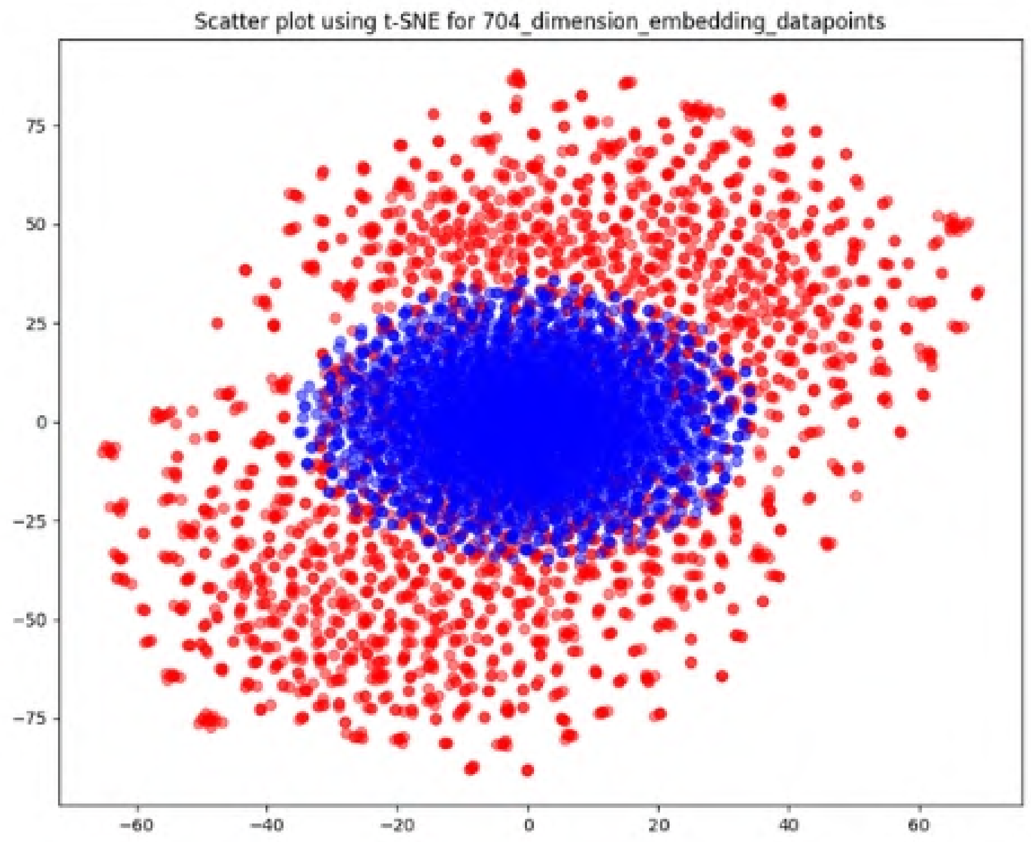

Try Pitch

# 704 Dimension Experiment

- The 704 generated embedding are lying in the centre of the latent space.
- This is simply because the distribution of the generated embedding still has not completely learned the distribution of the original embedding.
- The reason behind this is the increase in the dimensionality of the embedding, as it would require an increase in the number of training steps of the model.
- The generated 704 embeddings can be used to evaluate our performance on the real-world application as we used the embedding of dimension 704 and tested the performance on a text-to-speech IMS Toucan framework by the University of Stuttgart.
- Observed that every generated embedding is distinct.

# 704 Dimension Experiment




Try Pitch

# Results

- The difference in Mean and Variance between Original and Generated Embedding help in evaluating how
- well, the generated embedding is learnt.
- Ran all sets of experiments only for 500 steps to compare the convergence on a different set of embedding.

| Dimensions | Data points | Steps | Mean Difference | Variance Difference |
|---|---|---|---|---|
| 64 | 500 | 500 | $2.38e-07$ | $-4.4288$ |
| 64 | 1000 | 500 | $1.31e-06$ | $-10.16$ |
| 128 | 500 | 500 | $1.19e-07$ | $0.799$ |
| 128 | 1000 | 500 | $-1.86e-08$ | $0.389$ |
| 704 | 500 | 500 | $1.79e-07$ | $-0.7833$ |
| 704 | 1000 | 500 | $7.22e-16$ | $-1.4066$ |

**Table 1:** Difference in Mean and Variance between Original and Generated Embedding.

# Conclusion

- DDPM are successful at generating unseen speaker embedding
- Perform Controllability of Unseen Speaker Embedding with Eigenvector rotation which will lead to the generation of Speaker embedding bearing specific age, gender, etc

# References

[1] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In Helen Meng, Bo Xu, and Thomas Fang Zheng, editors, Interspeech 2020, pages 3830–3834. ISCA, 2020.
[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33:6840–6851, 2020.
[3] Florian Lux, Julia Koch, Antje Schweitzer, and Ngoc Thang Vu. The IMS Toucan system for the Blizzard Challenge 2021. In Proc. Blizzard Challenge Workshop, volume 2021. Speech Synthesis SIG, 2021.
[4] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In International Conference on Machine Learning, pages 8162–8171. PMLR, 2021.

# Thank you
# Questions?