


# Diffusion-Based Voice Conversion with **FAST** Maximum Likelihood Sampling Scheme



ICLR 2022

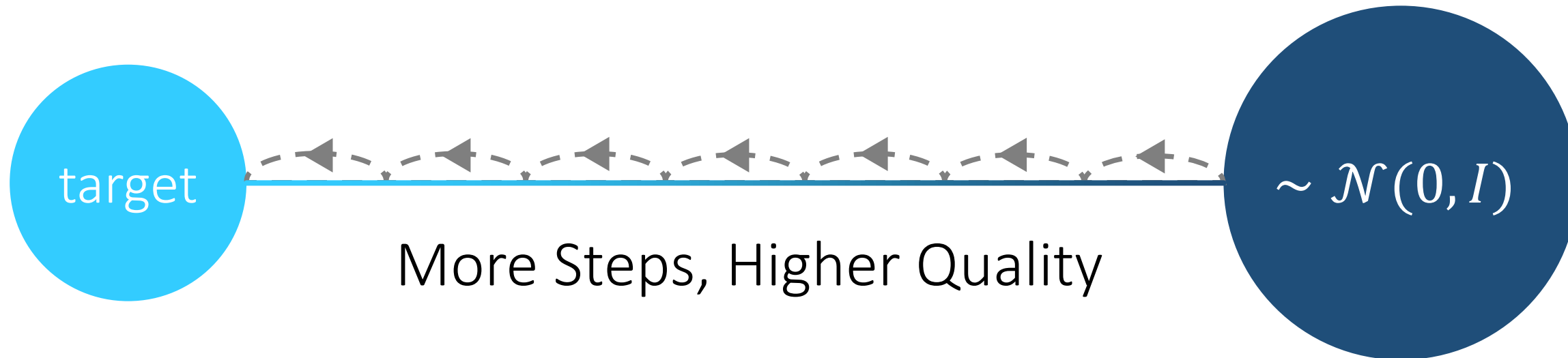
*Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova,  
Mikhail Sergeevich Kudinov, Jiansheng Wei*

# Why Use Diffusion Models?

- More Stable than GAN
- Higher Quality than VAE
- Easier to Design than Flow Models

But **Slower** than Them

Diffusion Steps

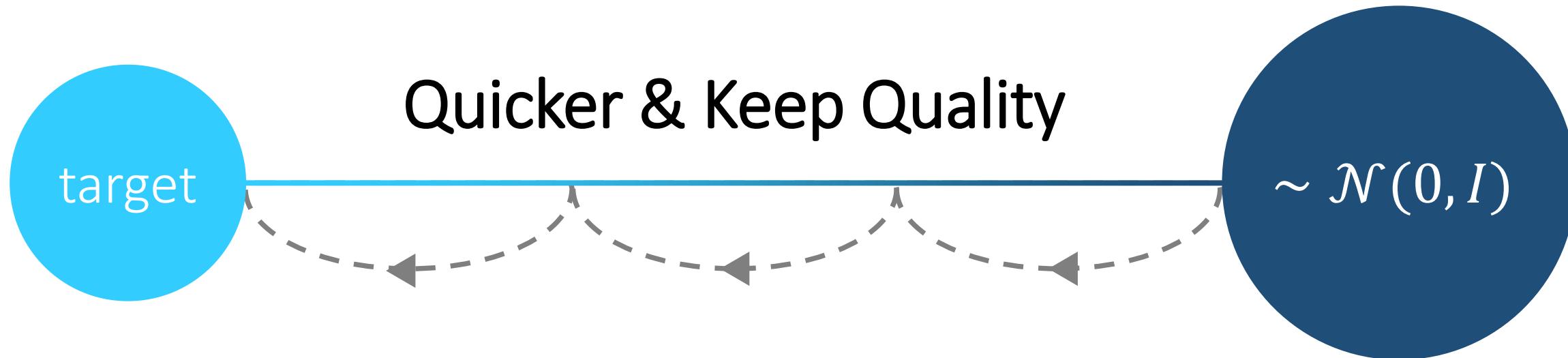


CIFAR10 ( $32 \times 32$ )					CelebA ( $64 \times 64$ )				
10	20	50	100	1000	10	20	50	100	1000
367.43	133.37	32.72	9.99	<b>3.17</b>	299.71	183.83	71.71	45.20	<b>3.26</b>

CIFAR10 and CelebA image generation measured in FID.

Image source: [Song et al., 2020](#)

Speed Up



Propose a **FAST** Sampling Scheme

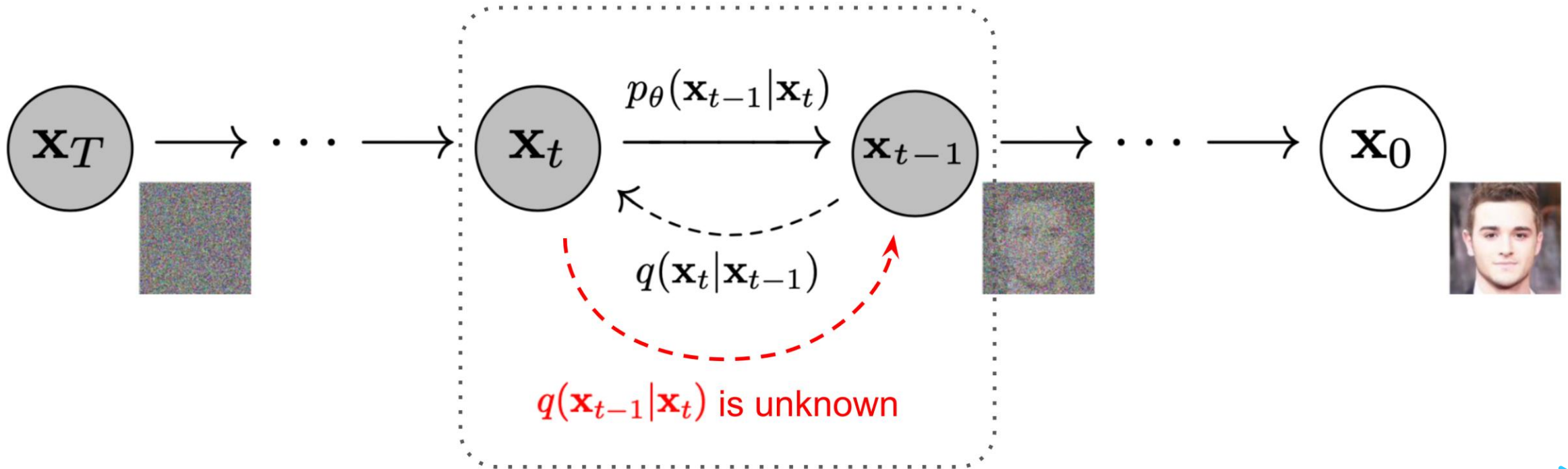
+

Average Voice Encoder

= SOTA Any to Any VC

# Forward Diffusion (Training)

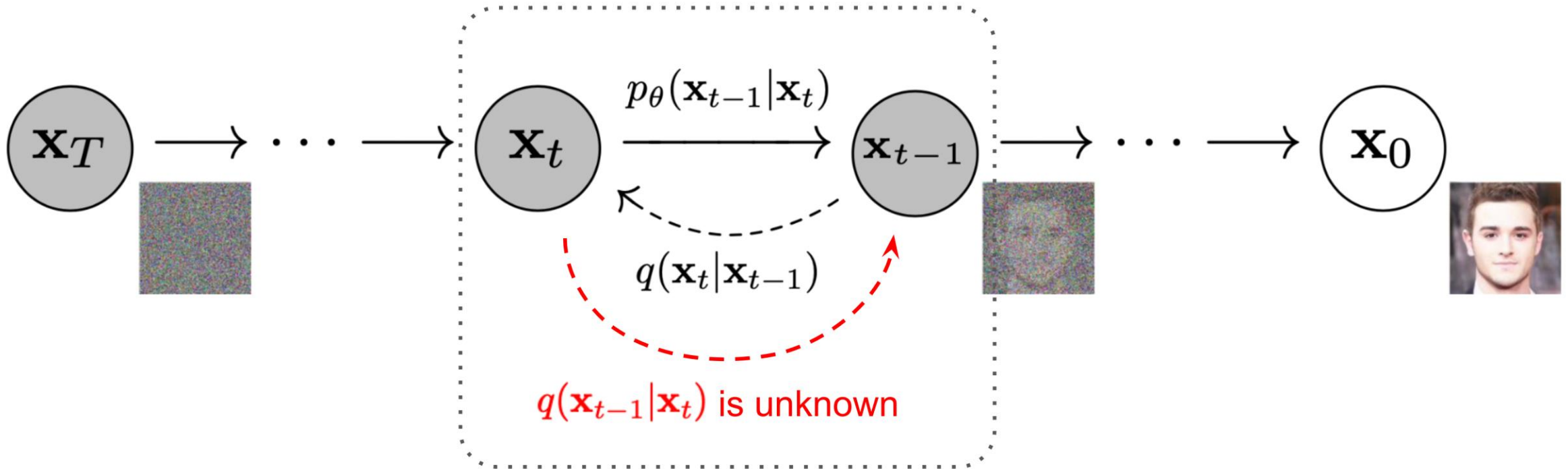
Use variational lower bound



# Reverse Diffusion (Inference)

# Forward Diffusion (Training)

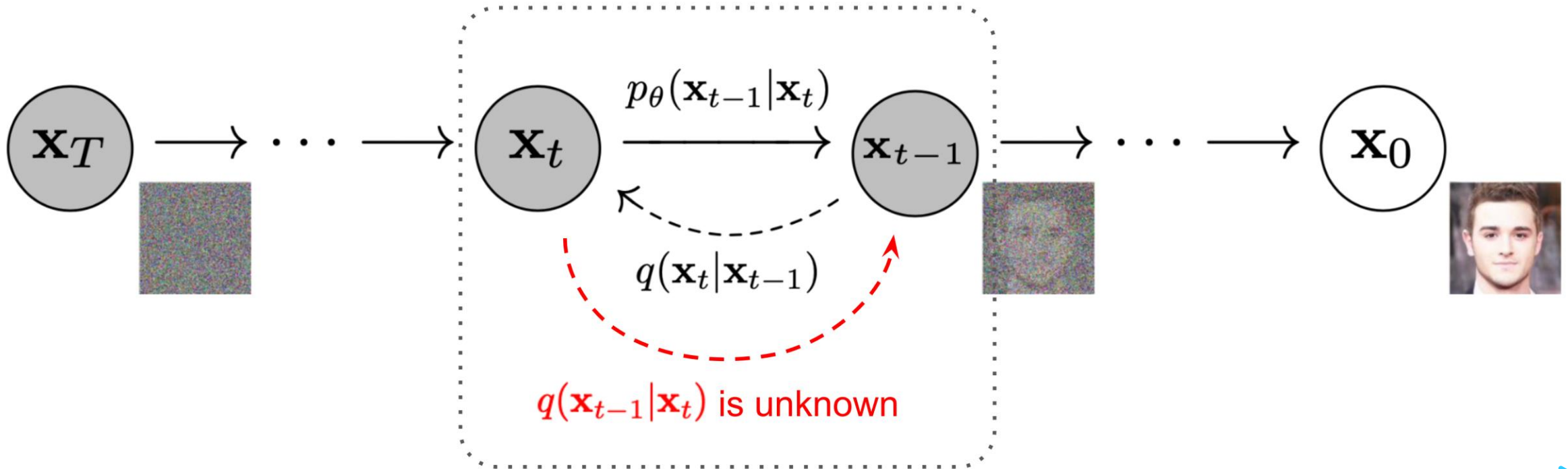
Use variational lower bound



Mix  $\mathbf{x}_0$  with noise to get  $\mathbf{x}_t$ , and train the model to estimate noise.

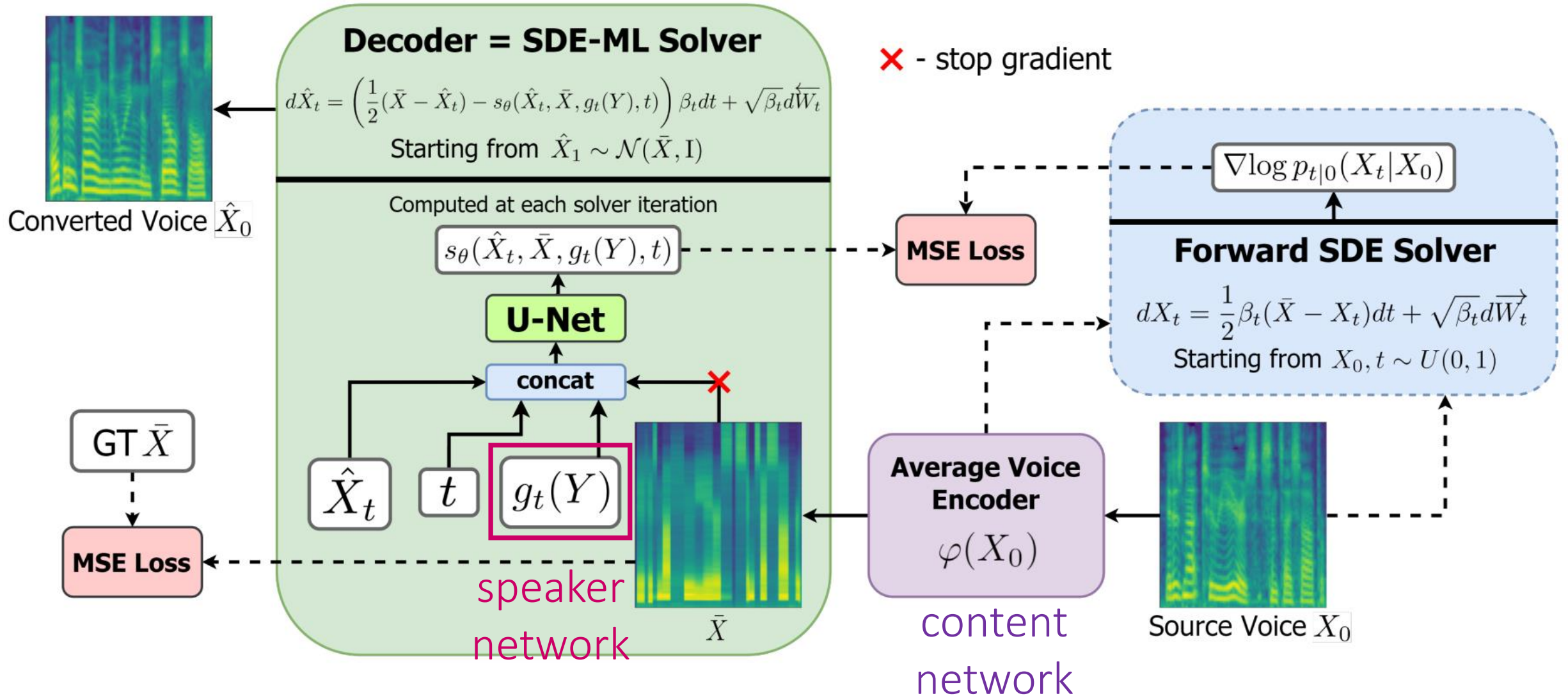
Reverse the diffusion process and sampling from it, you can generate real samples from Gaussian noise.

Use variational lower bound



Reverse Diffusion (Inference)

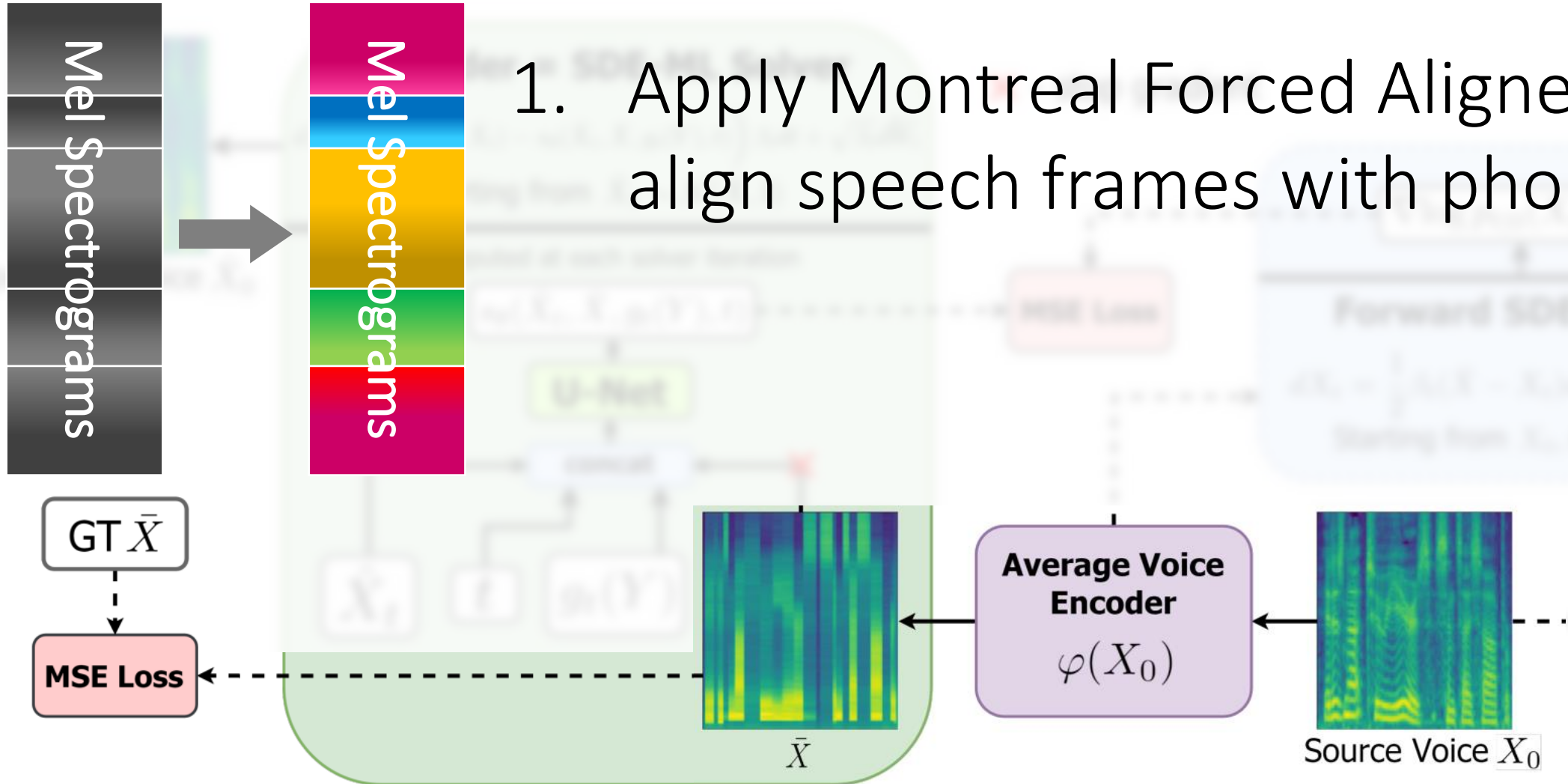
# Voice Conversion Diffusion Model





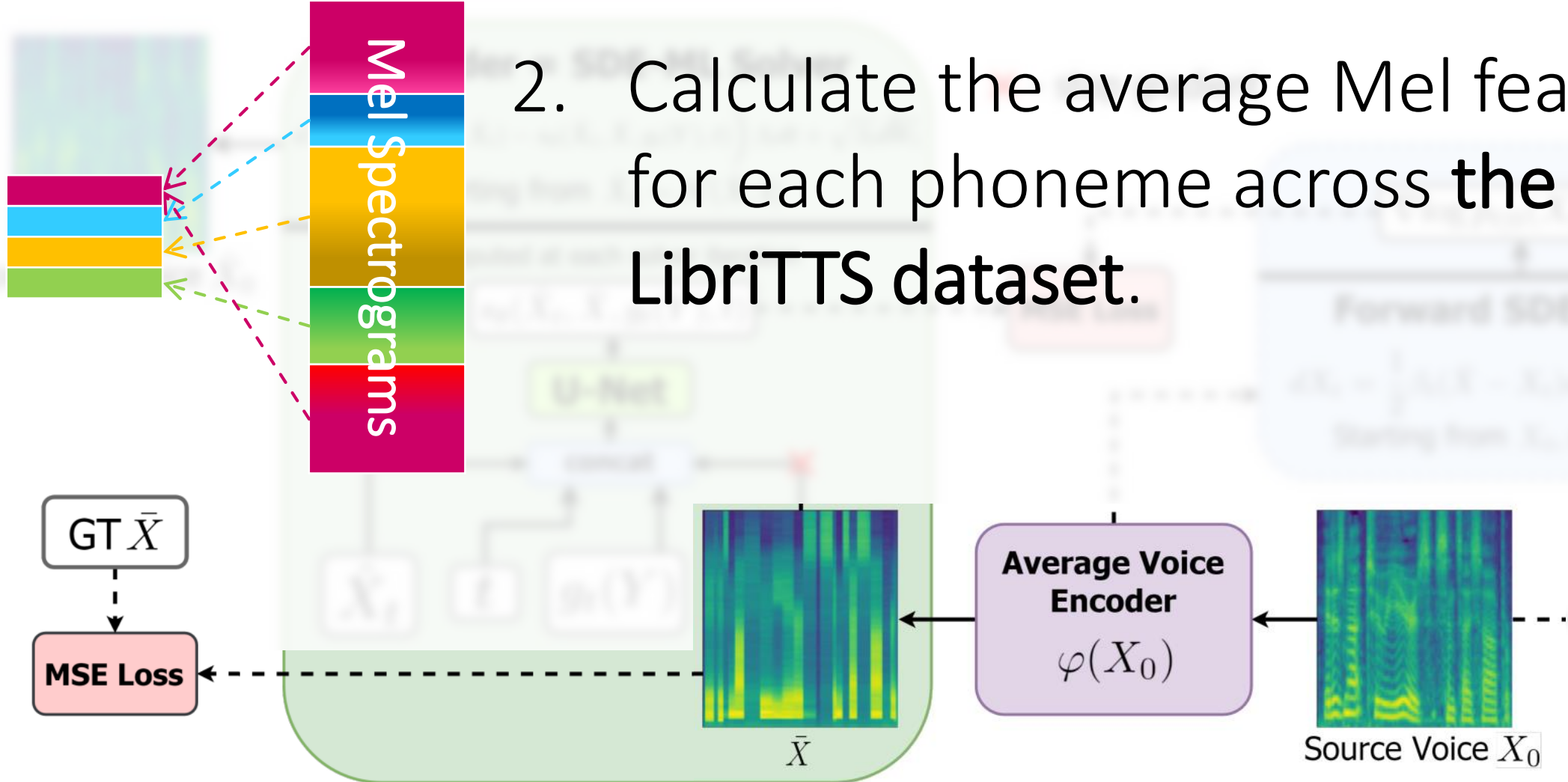
# Average Voice Encoder

1. Apply Montreal Forced Aligner to align speech frames with phonemes.



# Average Voice Encoder

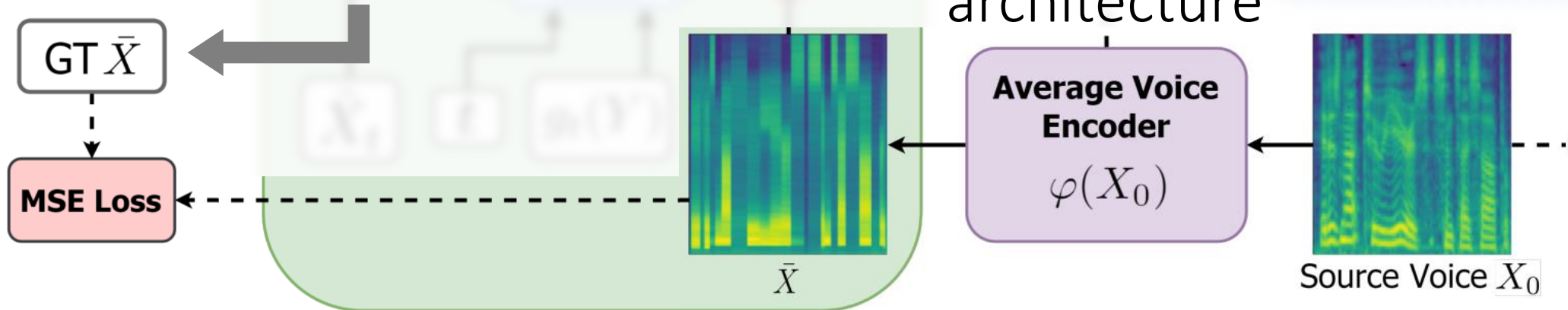
2. Calculate the average Mel feature for each phoneme across **the whole LibriTTS dataset**.



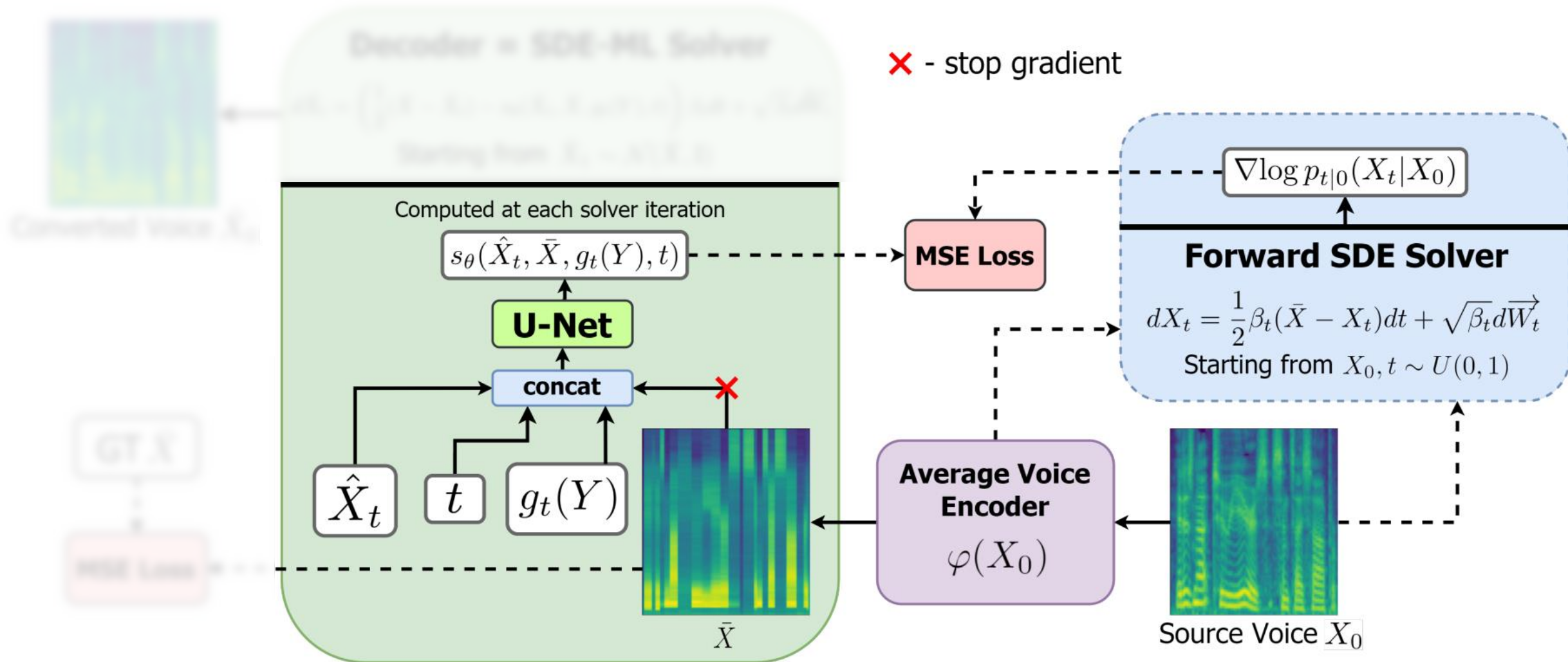
# Average Voice Encoder

3. Encoder is trained with “Average Voice” Mel Spectrogram as the target.

Transformer-based architecture



# Forward Diffusion



## Forward Diffusion: Sample $X_t$

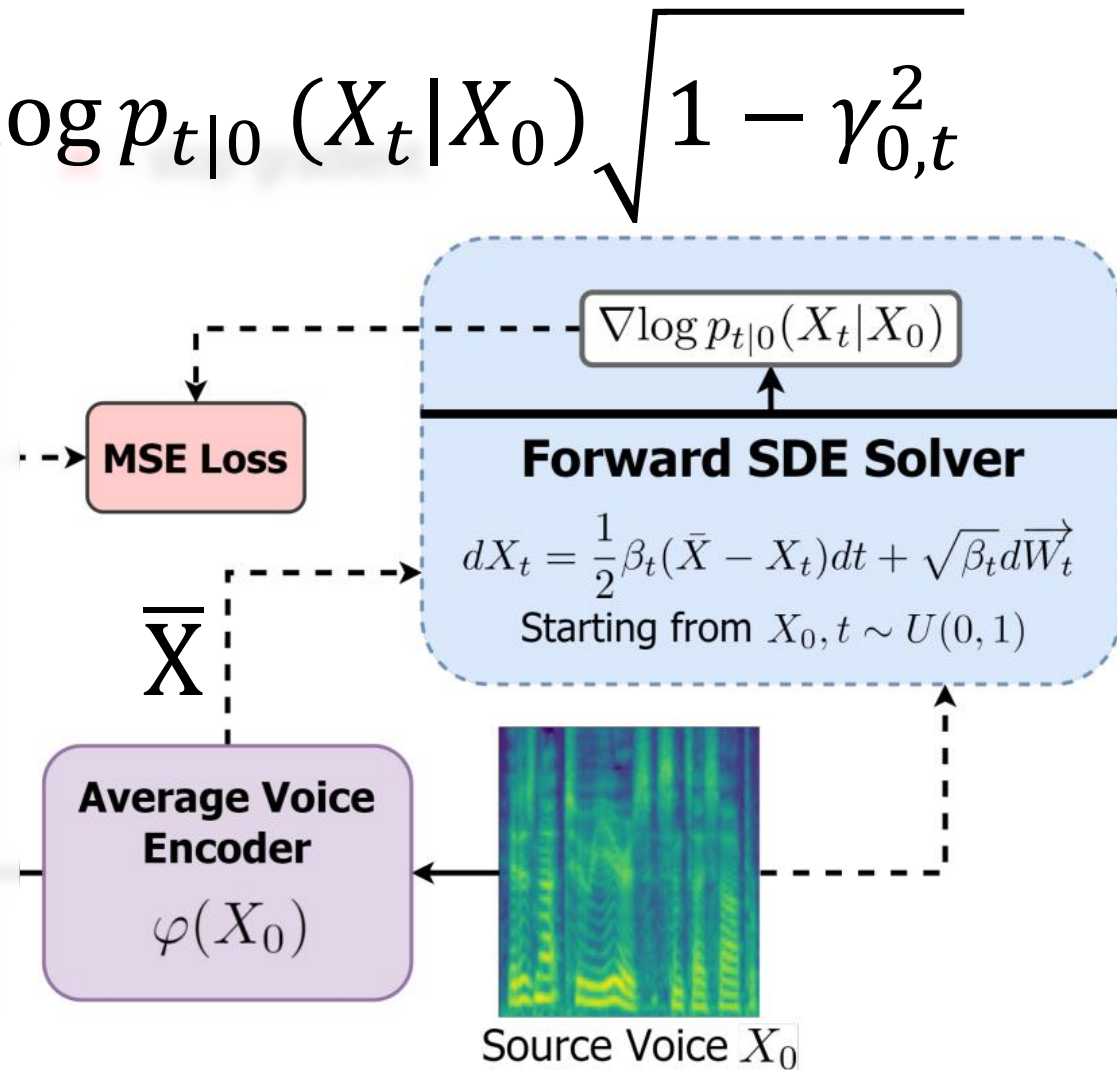
$$X_t = \gamma_{0,t}X_0 + (1 - \gamma_{0,t})\bar{X} + \nabla \log p_{t|0}(X_t|X_0)\sqrt{1 - \gamma_{0,t}^2}$$

$$\nabla \log p_{t|0}(X_t|X_0) \sim \mathcal{N}(0, I)$$

$$\gamma_{s,t} = e^{-\frac{1}{2} \int_s^t \beta_u du}$$

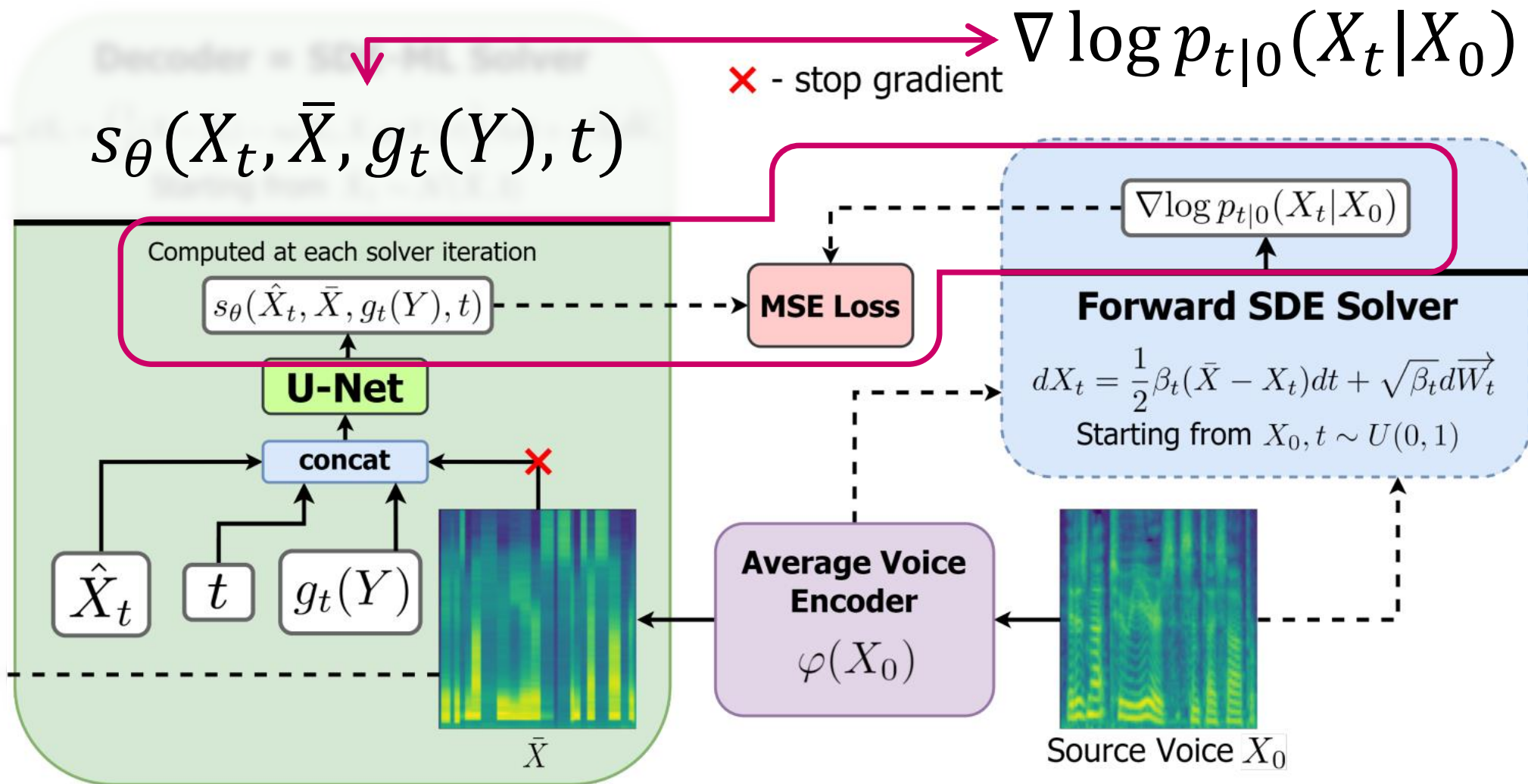
$$\beta_t = \beta_0 + t(\beta_1 - \beta_0)$$

Hyper parameters

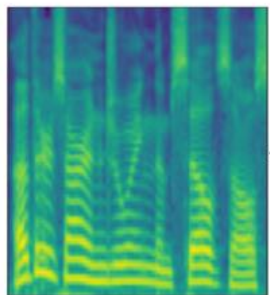




# Forward Diffusion: Loss



# Reverse Diffusion: Euler-Maruyama



Converted Voice  $\hat{X}_0$

**Decoder = SDE-ML Solver**

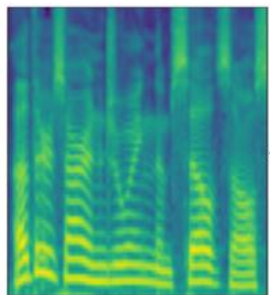
$$d\hat{X}_t = \left( \frac{1}{2}(\bar{X} - \hat{X}_t) - s_\theta(\hat{X}_t, \bar{X}, g_t(Y), t) \right) \beta_t dt + \sqrt{\beta_t} d\tilde{W}_t$$

Starting from  $\hat{X}_1 \sim \mathcal{N}(\bar{X}, I)$

$$\begin{aligned} \hat{X}_{t-h} &= \hat{X}_t + \hat{\sigma}_{t,h} \nabla \log p_{t|0}(X_t | X_0) \\ &\quad + \beta_t h \left( \left( \frac{1}{2} + \hat{\omega}_{t,h} \right) (\hat{X}_t - \bar{X}) + (1 + \hat{\omega}_{t,h}) s_\theta(X_t, \bar{X}, g_t(Y), t) \right) \end{aligned}$$

step size

# Reverse Diffusion: Maximum Likelihood



Converted Voice  $\hat{X}_0$

**Decoder = SDE-ML Solver**

$$d\hat{X}_t = \left( \frac{1}{2}(\bar{X} - \hat{X}_t) - s_\theta(\hat{X}_t, \bar{X}, g_t(Y), t) \right) \beta_t dt + \sqrt{\beta_t} d\hat{W}_t$$

Starting from  $\hat{X}_1 \sim \mathcal{N}(\bar{X}, I)$

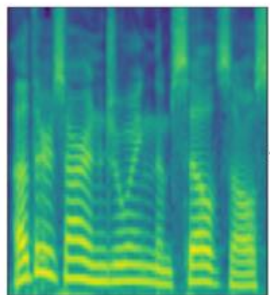
By Theorem 1.

$$\hat{\sigma}_{t,h} = \sigma_{t,h}^*, \hat{\omega}_{t,h} = \omega_{t,h}^*, \hat{\kappa}_{t,h} = \kappa_{t,h}^*$$

$$\begin{aligned} \hat{X}_{t-h} &= \hat{X}_t + \hat{\sigma}_{t,h} \nabla \log p_{t|0}(X_t | X_0) \\ &\quad + \beta_t \underset{\text{step size}}{h} \left( \left( \frac{1}{2} + \hat{\omega}_{t,h} \right) (\hat{X}_t - \bar{X}) + (1 + \hat{\kappa}_{t,h}) s_\theta(X_t, \bar{X}, g_t(Y), t) \right) \end{aligned}$$



# Reverse Diffusion: Maximum Likelihood



Converted Voice  $\hat{X}_0$

**Decoder = SDE-ML Solver**

$$d\hat{X}_t = \left( \frac{1}{2}(\bar{X} - \hat{X}_t) - s_\theta(\hat{X}_t, \bar{X}, g_t(Y), t) \right) \beta_t dt + \sqrt{\beta_t} d\hat{W}_t$$

Starting from  $\hat{X}_1 \sim \mathcal{N}(\bar{X}, I)$

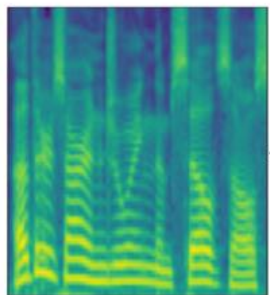
By Theorem 1.

$$\hat{\sigma}_{t,h} = \sigma_{t,h}^*, \quad \hat{\omega}_{t,h} = \omega_{t,h}^*, \quad \hat{\kappa}_{t,h} = \kappa_{t,h}^*$$

$$\kappa_{t,h}^* = \frac{\nu_{t-h,t}(1 - \gamma_{0,t}^2)}{\gamma_{0,t}\beta_t h} - 1, \quad \omega_{t,h}^* = \frac{\mu_{t-h,t} - 1}{\beta_t h} + \frac{1 + \kappa_{t,h}^*}{1 - \gamma_{0,t}^2} - \frac{1}{2},$$

$$(\sigma_{t,h}^*)^2 = \sigma_{t-h,t}^2 + \frac{1}{n} \nu_{t-h,t}^2 \mathbb{E}_{X_t} [\text{Tr}(\text{Var}(X_0|X_t))],$$

# Reverse Diffusion: Maximum Likelihood



Converted Voice  $\hat{X}_0$

**Decoder = SDE-ML Solver**

$$d\hat{X}_t = \left( \frac{1}{2}(\bar{X} - \hat{X}_t) - s_\theta(\hat{X}_t, \bar{X}, g_t(Y), t) \right) \beta_t dt + \sqrt{\beta_t} d\hat{W}_t$$

Starting from  $\hat{X}_1 \sim \mathcal{N}(\bar{X}, I)$

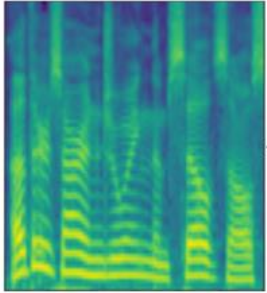
$$\mu_{s,t} = \gamma_{s,t} \frac{1 - \gamma_{0,s}^2}{1 - \gamma_{0,t}^2}, \quad \nu_{s,t} = \gamma_{0,s} \frac{1 - \gamma_{s,t}^2}{1 - \gamma_{0,t}^2}, \quad \sigma_{s,t}^2 = \frac{(1 - \gamma_{0,s}^2)(1 - \gamma_{s,t}^2)}{1 - \gamma_{0,t}^2},$$

$$\kappa_{t,h}^* = \frac{\nu_{t-h,t}(1 - \gamma_{0,t}^2)}{\gamma_{0,t}\beta_t h} - 1, \quad \omega_{t,h}^* = \frac{\mu_{t-h,t} - 1}{\beta_t h} + \frac{1 + \kappa_{t,h}^*}{1 - \gamma_{0,t}^2} - \frac{1}{2},$$

$$(\sigma_{t,h}^*)^2 = \sigma_{t-h,t}^2 + \frac{1}{n} \nu_{t-h,t}^2 \mathbb{E}_{\hat{X}_t} [\text{Tr}(\text{Var}(X_0 | X_t))],$$

Without in source code?

# Reverse Diffusion: Maximum Likelihood



Converted Voice  $\hat{X}_0$

**Decoder = SDE-ML Solver**

$$d\hat{X}_t = \left( \frac{1}{2}(\bar{X} - \hat{X}_t) - s_\theta(\hat{X}_t, \bar{X}, g_t(Y), t) \right) \beta_t dt + \sqrt{\beta_t} d\tilde{W}_t$$

Starting from  $\hat{X}_1 \sim \mathcal{N}(\bar{X}, I)$

$$\hat{X}_1 \sim \mathcal{N}(\bar{X}, I)$$

**for**  $i = 0$  **to**  $N - 1$  **do**

$$t \leftarrow i \times h$$

$$X'_{t-h} \leftarrow \hat{X}_t + \beta_t h \left( \left( \frac{1}{2} + \hat{\omega}_{t,h} \right) (\hat{X}_t - \bar{X}) + (1 + \hat{\kappa}_{t,h}) s_\theta(X_t, \bar{X}, g_t(Y), t) \right)$$

$$\nabla \log p_{t|0}(X_t | X_0) \sim \mathcal{N}(0, I)$$

$$\hat{X}_{t-h} \leftarrow X'_{t-h} + \hat{\sigma}_{t,h} \nabla \log p_{t|0}(X_t | X_0)$$

**return**  $\hat{X}_0$

# Speaker Conditional Analysis

Input types for speaker conditioning  $g_t(Y)$  compared in terms of speaker similarity.

	<i>Diff-LibriTTS</i>			<i>Diff-VCTK</i>		
	<i>d-only</i>	<i>wodyn</i>	<i>whole</i>	<i>d-only</i>	<i>wodyn</i>	<i>whole</i>
Most similar	27.0%	<b>38.0%</b>	34.1%	27.2%	<b>46.7%</b>	23.6%
Least similar	<b>28.9%</b>	29.3%	38.5%	25.3%	<b>23.9%</b>	48.6%

- d-only:  $Y$  = target Mel-spectrogram  $Y_0$
- wodyn:  $Y$  = **Noisy** target Mel-spectrogram  $Y_t$
- whole:  $Y = \{Y_t, Y_{0.5/15}, Y_{1.5/15}, \dots, Y_{14.5/15}\}$ , channel = 16

# Any-to-Any Voice Conversion

	VCTK test (9 speakers, 54 pairs)		Whole test (25 speakers, 350 pairs)	
	Naturalness	Similarity	Naturalness	Similarity
<i>AGAIN-VC</i>	$1.98 \pm 0.05$	$1.97 \pm 0.08$	$1.87 \pm 0.03$	$1.75 \pm 0.04$
<i>FragmentVC</i>	$2.20 \pm 0.06$	$2.45 \pm 0.09$	$1.91 \pm 0.03$	$1.93 \pm 0.04$
<i>VQMIVC</i>	$2.89 \pm 0.06$	$2.60 \pm 0.10$	$2.48 \pm 0.04$	$1.95 \pm 0.04$
<i>Diff-VCTK-ML-6</i>	<b><math>3.73 \pm 0.06</math></b>	$3.47 \pm 0.09$	$3.39 \pm 0.04$	<b><math>2.69 \pm 0.05</math></b>
<i>Diff-VCTK-ML-30</i>	<b><math>3.73 \pm 0.06</math></b>	<b><math>3.57 \pm 0.09</math></b>	<b><math>3.44 \pm 0.04</math></b>	<b><math>2.71 \pm 0.05</math></b>
<i>Ground truth</i>	$4.55 \pm 0.05$	$4.52 \pm 0.07$	$4.55 \pm 0.05$	$4.52 \pm 0.07$

Conv Auto Encoder  
Attention-based  
Vector Quantization

Train on VCTK, **100** speakers

All subjective human evaluation was carried out on Amazon Mechanical Turk.

# Any-to-Any Voice Conversion

	VCTK test (9 speakers, 54 pairs)		Whole test (25 speakers, 350 pairs)	
	Naturalness	Similarity	Naturalness	Similarity
<i>AGAIN-VC</i>	$1.98 \pm 0.05$	$1.97 \pm 0.08$	$1.87 \pm 0.03$	$1.75 \pm 0.04$
<i>FragmentVC</i>	$2.20 \pm 0.06$	$2.45 \pm 0.09$	$1.91 \pm 0.03$	$1.93 \pm 0.04$
<i>VQMIVC</i>	$2.89 \pm 0.06$	$2.60 \pm 0.10$	$2.48 \pm 0.04$	$1.95 \pm 0.04$
<i>Diff-VCTK-ML-6</i>	<b><math>3.73 \pm 0.06</math></b>	$3.47 \pm 0.09$	$3.39 \pm 0.04$	<b><math>2.69 \pm 0.05</math></b>
<i>Diff-VCTK-ML-30</i>	<b><math>3.73 \pm 0.06</math></b>	<b><math>3.57 \pm 0.09</math></b>	<b><math>3.44 \pm 0.04</math></b>	<b><math>2.71 \pm 0.05</math></b>
<i>Ground truth</i>	$4.55 \pm 0.05$	$4.52 \pm 0.07$	$4.55 \pm 0.05$	$4.52 \pm 0.07$

Conv Auto Encoder  
Attention-based  
Vector Quantization

Real-Time Factor on GPU (unknow model)

- 6 step: around **0.1**
- 30 step: around 0.5



# Any-to-Any Voice Conversion

Train on LibriTTS

approximately **1100** speakers.

	VCTK test (9 speakers, 54 pairs)		Whole test (25 speakers, 350 pairs)	
	Naturalness	Similarity	Naturalness	Similarity
<i>Diff-LibriTTS-EM-6</i>	$1.68 \pm 0.06$	$1.53 \pm 0.07$	$1.57 \pm 0.02$	$1.47 \pm 0.03$
<i>Diff-LibriTTS-PF-6</i>	$3.11 \pm 0.07$	$2.58 \pm 0.11$	$2.99 \pm 0.03$	$2.50 \pm 0.04$
<i>Diff-LibriTTS-ML-6</i>	$3.84 \pm 0.08$	$3.08 \pm 0.11$	$3.80 \pm 0.03$	$3.27 \pm 0.05$
<i>Diff-LibriTTS-ML-30</i>	<b><math>3.96 \pm 0.08</math></b>	$3.23 \pm 0.11$	<b><math>4.02 \pm 0.03</math></b>	<b><math>3.39 \pm 0.05</math></b>
<i>BNE-PPG-VC</i>	<b><math>3.95 \pm 0.08</math></b>	<b><math>3.27 \pm 0.12</math></b>	$3.83 \pm 0.03$	$3.03 \pm 0.05$

BEN-PPG-VC: combining a bottleneck feature extractor obtained from a phoneme recognizer with a seq2seq-based synthesis module.

# Any-to-Any Voice Conversion

The proposed maximum likelihood (ML) sampling scheme over other sampling methods for a small number of inference steps.

	VCTK test (9 speakers, 54 pairs)		Whole test (25 speakers, 350 pairs)	
	Naturalness	Similarity	Naturalness	Similarity
<i>Diff-LibriTTS-EM-6</i>	$1.68 \pm 0.06$	$1.53 \pm 0.07$	$1.57 \pm 0.02$	$1.47 \pm 0.03$
<i>Diff-LibriTTS-PF-6</i>	$3.11 \pm 0.07$	$2.58 \pm 0.11$	$2.99 \pm 0.03$	$2.50 \pm 0.04$
<i>Diff-LibriTTS-ML-6</i>	$3.84 \pm 0.08$	$3.08 \pm 0.11$	$3.80 \pm 0.03$	$3.27 \pm 0.05$
<i>Diff-LibriTTS-ML-30</i>	<b><math>3.96 \pm 0.08</math></b>	$3.23 \pm 0.11$	<b><math>4.02 \pm 0.03</math></b>	<b><math>3.39 \pm 0.05</math></b>
<i>BNE-PPG-VC</i>	<b><math>3.95 \pm 0.08</math></b>	<b><math>3.27 \pm 0.12</math></b>	$3.83 \pm 0.03$	$3.03 \pm 0.05$

BEN-PPG-VC: combining a bottleneck feature extractor obtained from a phoneme recognizer with a seq2seq-based synthesis module.



# Maximum Likelihood Sampling

Euler-Maruyama



Probability Flow



Maximum Likelihood



CIFAR-10 images randomly sampled from VP DPM by running 10 reverse diffusion steps.

## Conclusion

- Average Voice Encoder  
a new disentanglement method.
- Diffusion-based Decoder  
achieve good results both in terms of similarity and naturalness.
- Novel Sampling Scheme  
High-quality results in just a few steps.